1991年6月1日発行（毎月1回1日発行）第10巻6号通巻110号　昭和58年11月2日第三種郵便物認可

PERSONAL COMPUTER MAGAZINE for MZ, X1, and X68000

特集　初心者のための環境構成術

6

1991

創刊9周年
特別記念号

X68000に98用マウスを/アンケート結果大分析大会その1
マルチウィンドウシステムSystem-7C/愛読者特大プレゼント
全機種共通システムS-OS 6周年記念 Small-C処理系の移植
新連載 響子 in CGわ～るど/CARD PRO-68K Ver.2.0

オー／エックス
特別定価600円

資料請求券
X68000
Oh!X
6係

いまクロック16MHzの俊才、「エクシヴィ」のデビューで5年に及ぶ68000CPUへの探求は、ひとつの結論を得ようとしています。極めたといえば言い過ぎでしょうが、事の深淵に迫ろうと努力するもののみに与えられる深い充足を、私たちスタッフは、これまでX68000を支えていただいたユーザー、ソフトハウス、ハードベンダー諸兄とともに味わいたい心境です。徹底したこだわりと、それを裏付けるアドバンストテクノロジー、世間の逆風を揚力にしてしまう、それなりの魅力と知性を背景として備えたX68000が、パーソナルコンピュータに新しいジャンルを切り拓いてきた歩みは、ご存じの通りです。現在のマルチメディア環境を開発当初から想定していた先見性。一言でいえばクリエイティブマインドということでしょうが、そのグラフィックアビリティ、映像統合コンセプト、サンプリング音源、ウィンドウ環境、そうした単に、とはいえ凄いスペックさえ超えたところにX68000の付加価値は存在します。アプリケーションを走らせるだけのブラックボックス化した、あるいは文房具としてのマシン、それはそれで異論はないのですが、本来的にパーソナルコンピュータがもつ可能性を育む、いわば創造性という観点から物足りなさを覚えることも事実です。X68000は、ある意味ではたいへんな異端児かも知れません。しかし世間から見たその"異能"は、私たちが考えるパーソナルコンピュータとしてはまさにスタンダードに他なりません。いつも新鮮な感動がある、驚きがある。新しい発見がある。"センス"の違いはスペックをも超えて使う人に訴えかける、敢えて68000CPUに執着してきた理由もここにあります。ワークステーションとしての成熟、先見性、創造性の具現化、ユーザーインターフェイスの追求。X68000の進化の過程はここに凝縮されています。

——新しい「エクシヴィ」がこのコンセプトをどう発展させたか、ご体感ください。

瞬速16MHz、エクシヴィ登場。

16MHzクロック68000搭載：OSの高速化、アートワークをパワフルにサポートするクロック周波数16MHzの68000CPUを搭載。クリエイティブワークステーションにふさわしいシステムパフォーマンスを実現しました。

SX-WINDOW ver1.1搭載：CPUのクロックアップと合わせ、大幅な処理速度の向上を実現。操作性を一段と高めたニューバージョンです。多機能・高速の強力エディタを搭載。文字選択・外字作成ツールも装備して、スムーズな日本語入力環境をサポート。またプリンタデバイスドライバを搭載し、多彩な印字指定が可能です。もちろん、こうした新しい環境がすべてのX68000で享受できることは言うまでもありません。そして待望のウィンドウアプリケーションもリリースされはじめています。

高密度メモリ拡張環境：メインメモリは標準で2Mバイト、本体内部のメイン基板上に6Mバイト増設でき、I/Oスロットを使用せず最大8Mバイトの高速メモリアクセスを実現。さらにI/Oスロットへの増設を含め最大12Mバイトまで拡張できます。数値演算プロセッサも本体内に取り付けられます。

※2MB増設メモリ（ボード型）CZ-6BE2A　標準価格59,800円（税別）、2MB増設メモリ（チップ型）CZ-6BE2B　標準価格54,800円（税別）、数値演算プロセッサ（チップ型）CZ-6BP2　標準価格45,800円（税別）を使用。（すべて別売）

●大容量メディア対応、世界標準SCSIインターフェイス標準装備●X68000シリーズとフルコンパチブル設計●高品位なチタンブラックのニューデザインマンハッタンシェイプ●81MバイトSCSI仕様HDD搭載（CZ-644C）/内蔵可能（CZ-634C）●1024×1024ドットの実画面エリアを装備した高解像度表示（最大表示エリア768×512ドット・65,536色中16色表示）、65,536色同時表示（512×512ドット時）、先駆の高解像度自然色グラフィック●AD PCM、ステレオ8オクターブ8重和音FM音源搭載●オートロード・オートイジェクトの1Mバイト5インチFDD2基搭載●マウス・トラックボール標準装備

●写真はCZ-644C-TNとCZ-613D-TN。

本体＋キーボード＋マウス・トラックボール
CZ-634C-TN（チタンブラック）　標準価格368,000円（税別）
81MB HDタイプ　CZ-644C-TN（チタンブラック）　標準価格518,000円（税別）

SUPER 本体＋キーボード＋マウス・トラックボール
CZ-604C-TN（チタンブラック）　標準価格348,000円（税別）
81MB HDタイプCZ-623C-TN（チタンブラック）　標準価格498,000円（税別）

PROII 本体＋キーボード＋マウス
CZ-653C-BK（ブラック）・-GY（グレー）　標準価格285,000円（税別）
40MB HDタイプCZ-663C-BK（ブラック）・-GY（グレー）　標準価格395,000円（税別）

●15型カラーディスプレイテレビ（ドットピッチ0.39mm）　CZ-602D-BK（ブラック）・-GY（グレー）※　標準価格99,800円（チルトスタンド同梱・税別）
●15型カラーディスプレイテレビ（ドットピッチ0.39mm）　CZ-605D-BK（ブラック）・-GY（グレー）　標準価格115,000円（スピーカー2個/チルトスタンド同梱・税別）
●15型カラーディスプレイテレビ（ドットピッチ0.31mm）　CZ-613D-TN（チタンブラック）・-BK（ブラック）・-GY（グレー）　標準価格135,000円（スピーカー2個/チルトスタンド同梱・税別）
●14型カラーディスプレイ（ドットピッチ0.31mm）　CZ-603D-BK（ブラック）・-GY（グレー）　標準価格84,800円（チルトスタンド同梱・税別）
●14型カラーディスプレイ（ドットピッチ0.31mm）　CZ-604D-BK（ブラック）・-GY（グレー）　標準価格94,800円（スピーカー2個/チルトスタンド同梱・税別）
●14型カラーディスプレイ（ドットピッチ0.31mm）　CZ-606D-TN（チタンブラック）・-BK（ブラック）・-GY（グレー）　標準価格79,800円（チルトスタンド同梱・税別）
●21型カラーディスプレイ（ドットピッチ0.52mm）　CU-21HD-BK（ブラック）　標準価格148,000円（スピーカー2個同梱・税別）

※印の商品は在庫僅少です。

68買ったらEXEクラブに入ろう！

本体同梱の入会申込ハガキを送るだけで、無料入会。3つのメリット！
メリット1：会員No.入りオリジナル**会員証電卓**がもらえる。
メリット2：各種フェアご優待・イベントご案内等、数々の特典あり。
メリット3：X68000の活用情報が手に入る「**EXEおみこし活動**」に参加できる。
※「申込ハガキをなくしてしまった」という方は、「おみこし活動隊」☎(06)886-0354までお電話ください。

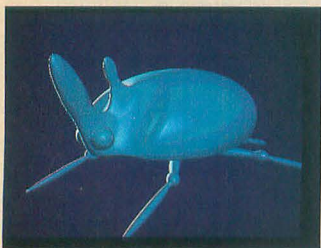
EXEおみこし活動とは？
コミュニケーションペーパー「おみこしPRESS」を通じて会員同士が情報を交換、どこまでもX68000を使いこなして盛り上がってしまおう！というのが、その目的。68へのラブコール、会員独自のテクニック・活用法など、あなたの68自慢を「おみこし活動隊」までどうぞ。会員メッセージは随時「おみこしPRESS」に掲載します。

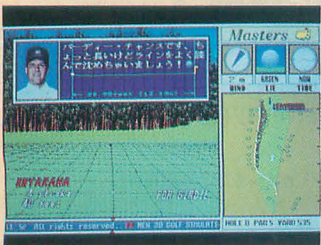
●お問い合わせは…
シャープ株式会社
電子機器事業本部システム機器営業部
〒545 大阪市阿倍野区長池町22番22号☎(06)621-1221（大代表）
電子機器事業本部液晶映像システム事業部第2商品企画部
〒162 東京都新宿区市谷八幡町８番地☎(03)3260-1161（大代表）

Oh!X

CONT

特集 初心者のための環境構成術

System-7C

C-TRACE TP+

パロディウスだ!

遥かなるオーガスタ

PC-9801用マウスを使う

〈スタッフ〉
●編集長/前田 徹 ●副編集長/植木章夫 ●編集/岡崎栄子 浅井研二 山田純二 ●協力/有田隆也 中森 章 林 一樹 吉田幸一 華門真人 毛内俊行 吉田賢司 影山裕昭 古村 聡 村田敏幸 丹 明彦 三沢和彦 長沢淳博 宮島 靖 金子俊一 浦川博之 石上達也 ●カメラ/杉山和美 ●イラスト/永沢しげる 山田晴久 小栗由香 ●アートディレクター/島村勝頼 ●レイアウト/元木昌子 AD GREEN ●校正/グループごじら

表紙絵：須藤　牧人

E　N　T　S

1991 JUN.
6

■広告目次

SHARP

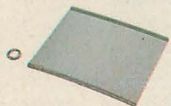
システムパフォーマンスを実証する多彩なペリフェラル。

ディスプレイ関連

カラーディスプレイテレビ

15型カラーディスプレイテレビ
★CZ-602D-BK
★CZ-602D-GY
標準価格99,800円(税別)
(チルトスタンド同梱)

15型カラーディスプレイテレビ
CZ-605D-BK・-GY
標準価格115,000円(税別)
(スピーカー2個・チルトスタンド同梱)

15型カラーディスプレイテレビ
CZ-613D-TN・-BK・-GY
標準価格135,000円(税別)
(スピーカー2個・チルトスタンド同梱)

CRTフィルター

高性能CRTフィルター
BF-68PRO
標準価格19,800円(税別)
(14/15型用)

カラーディスプレイ

NEW
14型カラーディスプレイ
CZ-606D-TN・BK・-GY
標準価格79,800円(税別)
(チルトスタンド同梱)

14型カラーディスプレイ
CZ-604D-BK・-GY
標準価格94,800円(税別)
(スピーカー2個・チルトスタンド同梱)

21型カラーディスプレイ
CU-21HD
標準価格148,000円(税別)
(スピーカー2個同梱)

チューナー

RGBシステムチューナー
CZ-6TU-BK・-GY
標準価格33,100円(税別)
(リモコン付)

アートツール

画像入力

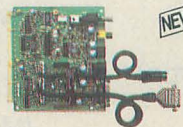
カラーイメージスキャナ※1
★CZ-8NS1
標準価格188,000円(税別)

NEW
カラーイメージスキャナ※1
JX-220X
標準価格168,000円(税別)
※RS-232C/パラレルインターフェイス標準装備

スキャナ用パラレルボード
★CZ-6BN1
標準価格29,800円(税別)

映像入力

カラーイメージユニット※2
CZ-6VT1-BK
CZ-6VT1
標準価格69,800円(税別)

映像出力

NEW
ビデオボード※3
CZ-6BV1
標準価格21,000円(税別)

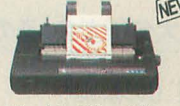
プリンタ

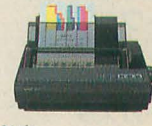
熱転写カラープリンタ

NEW
48ドット
熱転写カラー漢字プリンタ
CZ-8PC5-BK
標準価格96,800円(税別)

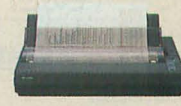
カラービデオプリンタ

カラービデオプリンタ
★CZ-6PV1
標準価格198,000円(税別)
(信号ケーブル同梱)

カラーイメージジェット

NEW
カラーイメージジェット※4
IO-735X-B
標準価格248,000円(税別)
(信号ケーブル別売)
※グレータイプのIO-735Xもあります。

カラードットプリンタ

24ピン
カラー漢字プリンタ(80桁)
CZ-8PG1
標準価格130,000円(税別)
(信号ケーブル同梱)

24ピン
カラー漢字プリンタ(136桁)
CZ-8PG2
標準価格160,000円(税別)
(信号ケーブル同梱)

ドットプリンタ

24ピン漢字プリンタ(136桁)
CZ-8PK10
標準価格97,800円(税別)
(信号ケーブル同梱)

ファイル

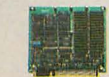
光磁気ディスク

NEW
光磁気ディスクユニット※5
(594MB)
CZ-6MO1
標準価格450,000円(税別)
(SCSIケーブル同梱)
※光磁気ディスクカートリッジは別売です。別売のJY-701MPA 標準価格30,000円(税別)をご使用ください。

ハードディスク

増設用ハードディスクドライブ(40MB)
(CZ-602C/603C/652C/653C内蔵用)
★CZ-64H※
標準価格120,000円(税別)
(取付費別)

NEW
増設用ハードディスクドライブ(81MB)
(CZ-604C/634C内蔵用)
CZ-68H※
標準価格160,000円(税別)
(取付費別)
※取付に関してはシャープお客様ご相談窓口にてご相談ください。

ハードディスクユニット(20MB)
★CZ-620H
標準価格178,000円(税別)
※604C/623C/634C/644Cでは使用できません。

※1 ご使用に際しては、カラーイメージスキャナCZ-8NS1、JX-220Xに同梱のRS-232Cケーブルで接続するか、より高速のパラレルデータ伝送を行う場合、別売のスキャナ用パラレルボードCZ-6BN1標準価格29,800円(税別)で接続してください。※2 テレビチューナーを内蔵していないディスプレイをご使用の場合は、RGBシステムチューナーCZ-6TU(別売)が必要です。※3 ビデオ出力は15.75kHzテレビ標準信号です。また、拡張I/Oスロットは2スロット使用します。※4 別売の信号ケーブルIO-73CX標準価格5,500円(税別)で接続してください。※5 CZ-600C、601C、602C、603C、611C、612C、613C、652C、653C、662C、663Cにご使用の場合は、別売のSCSIボード(CZ-6BS1)が必要です。また、X68000用 OS Human 68k ver 2.0以上にてご使用ください。(光磁気ディスクカートリッジは別売のJY-701MPA 標準価格30,000円(税別)をご使用ください。) ※6 ご使用に際しては、あらかじめ別売の1MB増設RAMボード CZ-6BE1 標準価格 35,000円(税別・

資料請求券
X68000ペリフェラル
Oh!X
6俵

シャープ株式会社 ●お問い合わせは…シャープ㈱電子機器事業本部システム機器営業部 〒545 大阪市阿倍野区長池町22番22号☎(06)621-1221(大代表

SX-WORKS

ウィンドウアプリケーションのファーストリリース。

X68000にふさわしい環境としてこだわり続けてきた
「ウィンドウ環境」が、いよいよ始動します。
操作環境、快適さ、グラフィカルユーザーインターフェイスを推進するSX-WINDOWの真価をご体験ください。
待望のウィンドウアプリケーション、あの感動が甦ります。

● SX-WINDOW対応グラフィックツール

CZ-263GW

5月発売予定

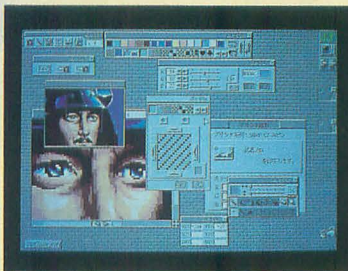
同時に複数のウィンドウを開いて編集できるSX-WINDOW対応初のグラフィックツール。65,536色中16色のカラー編集に加えて、わかりやすいアイコン表示。各ウィンドウ間のデータのやりとり、他のSX-WINDOWアプリケーションとのイメージデータの相互利用も可能です。

※本ソフトの動作には、メインメモリ2MBおよびSX-WINDOW ver1.1が必要です。

● SX-WINDOW ver1.1(CZ-278SS)5月発売予定

※SX-WINDOW ver1.0(CZ-259SS)を既にお持ちの方には有償バージョンアップを行います。

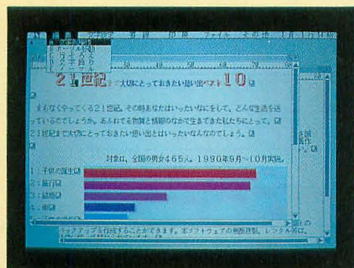
● Easydraw SX-68Kも開発中、ご期待ください!

資料請求券
CZソフト
Oh!X
6係

NEW RELEASE

ウィンドウでWYSIWYG編集。
カラーグラフィック、高速テキストモード。

マルチワープロ PRO-68K
Multiword

CZ-225BS　標準価格32,000円(税別)

WYSIWYGな編集が行えるウィンドウモードと素早い編集が行えるテキストモードをサポート。グラフィックを文章中に自由にレイアウトできます。また同一文章での複数の改行幅指定を可能にするなど多彩な機能を装備。レーザープリンタ、カラー印字(8色)の高品位プリントアウトも可能です。

※メインメモリ2MB必要です。

パソコン通信も、エディタも。
【メモリ常駐型】の優れもの。

Teleportion PRO-68K

CZ-258BS　標準価格22,800円(税別)

他のソフトウェアを実行中でも任意に呼び出して使える メモリ常駐型 のソフトウェアです。パソコン通信/エディタ/カレンダー/スケジュール/住所録/メモ帳/関数電卓の機能を文具感覚でお使いいただけます。「シャープ電子手帳」のデータをX68000で簡単に入力・編集することができます。

※メインメモリ2MB必要です。※StationeryPRO-68K(CZ-240BS)をお持ちの方には有償バージョンアップサービスを行います。

処理速度の高速化を実現。
Zeit日本語ベクトルフォントをサポート。

CZ-265HS　7月発売予定

オリジナリティを活かせるホームプロダクティビリティツールです。処理速度の高速化を実現。カセットレーベル、カレンダー作成に対応したほか、モノクロデータの編集などグラフィックエディタを強化し高機能テキストエディタを内蔵しました。Zeit日本語ベクトルフォントも使用可能です。

※メインメモリ2MB必要です。※NEW Print Shop PRO-68K(CZ-221HS)をお持ちの方には有償バージョンアップサービスを行います。

●お問い合わせは…シャープ㈱電子機器事業本部液晶映像システム事業部第2商品企画部 〒162 東京都新宿区市谷八幡町8番地 ☎(03)3260-1161(大代表)へ。 シャープ株式会社

SHARP

X68000 XVIデビュー記念キャンペーン

「夢、創ります。山下章氏プロデュース」

第1回全日本X68000
芸術祭開催

X68000アイドル山下章氏司会、進行による
ユーザー参加型作品コンテスト

■主催：シャープ株式会社 電子機器事業本部 システム機器営業部
■共催：シャープエレクトロニクス販売株式会社各統轄営業部
東京中央シャープ販売㈱・浪速シャープ電機㈱・
沖縄シャープ電機㈱
■協賛：出版社・ソフトハウス・サードパーティ・主要販売店

7月からの地区予選に始まる全国規模のオリジナルソフトウェア・作品コンテストです。日頃の腕試しに、ぜひ力作をお寄せください。全国のパソコンユーザーを魅了する芸術祭グランプリをめざして、あなたの自信作はどこまで勝ち進めるか？乞う御期待！

〈作品応募要項〉■**作品基準**：パーソナルコンピュータ（メーカー、機種を問わず）で制作した、オリジナル未発表のプログラム、グラフィックス、コンピュータ・ミュージック等であること。なお応募作品（制作に使用したアプリケーション・ソフト等以外の部分）の著作権は、すべてシャープ㈱に帰属します。■**部門**：①**ゲーム部門**、②**ミュージック部門**（自作の曲／一般曲・ゲームミュージックのアレンジ等、MIDI使用も可。）③**グラフィックス部門**（Z'sSTAFF PRO-68K、DOGA等のツールを使用して描いたものなど画面上に表示されるグラフィックスなら何でも可。）④**その他部門**：（ユーティリティ／一発ギャグ／パフォーマンス／ビジネス利用／その他）※応募は、1部門につき1人1作。1人複数部門応募は可。また団体制作も可。■**応募資格**：各予選ブロックの地域の住人であること。■**応募方法**：プログラム・ディスクに住所／氏名／年齢／職業（学校名・学年）／電話番号／開発に要した期間／開発に使用・利用したツール名／セールスポイント／取り扱い上の注意／動作に必要とする特殊機材を添え、各地区の応募先まで郵送してください。締め切りはその地区の地区大会開催日の2週間前（必着）です。■**賞・賞品**：〈地区予選〉●各地区大会大賞（1点）トロフィー、賞状、副賞●入選（首都圏3点、近畿2点、中部・九州2点、他地区なし）賞状、副賞●協賛各社賞・賞状、副賞〈全国大会〉●第1回全日本X68000芸術祭グランプリ（1点）トロフィー、賞状、副賞：光磁気ディスクユニット（CZ-6MO1）ペアでの海外旅行（旅行クーポン）●ゲーム・ミュージック・グラフィック等各部門賞・賞状、副賞●協賛各社賞・賞状、副賞

※詳細は店頭のチラシをご覧ください。

開催月（予定）	開催地	対象都道府県	応募・問い合わせ先
7月	四国地区（高松）	徳島・香川・愛媛・高知	〒760 高松市朝日町6-2-8 シャープエレクトロニクス販売㈱ 四国統轄（営） パソコン担当、辻井部長・細川係長 ☎0878-23-4860㈹
8月	北海道地区（札幌）	北海道	〒063 札幌市西区二十四軒1条7-3-17 シャープエレクトロニクス販売㈱ 北海道統轄（営） 商品営業部、長谷田 ☎011-642-8111㈹
8月	中国地区（広島）	鳥取・島根・岡山・広島・山口	〒731-01 広島市安佐南区西原2-13-4 シャープエレクトロニクス販売㈱ 中国統轄（営） パソコン担当、青木部長・石井 ☎082-874-2282㈹
9月	東北地区（仙台）	青森・山形・岩手・福島・宮城・秋田	〒983 仙台市若林区卸町東3-1-27 シャープエレクトロニクス販売㈱ 東北統轄（営） パソコン担当、岡本部長・阿部課長 ☎022-288-9111㈹
9月	北関東地区（宇都宮）	茨城・群馬・栃木	〒320 宇都宮市不動前4-2-41 シャープエレクトロニクス販売㈱ 北関東統轄（営） パソコン担当、岩田部長・川俣係長 ☎0286-35-1151㈹
10月	神奈川地区（横浜）	神奈川	〒235 横浜市磯子区中原1-2-23 シャープエレクトロニクス販売㈱ 神奈川統轄（営） パソコン担当、常次部長 ☎045-753-5501㈹
10月	中部地区（名古屋）	静岡・愛知・長野・岐阜・三重	〒454 名古屋市中川区山王3-5-5 シャープエレクトロニクス販売㈱ 中部統轄（営） パソコン担当、山口課長 ☎052-323-5111㈹
11月	北陸地区（金沢）	富山・石川・福井	〒921 石川県石川郡野々市町字御経塚町1096-1 シャープエレクトロニクス販売㈱ 北陸統轄（営） パソコン担当、小林 ☎0762-49-1181㈹
11月	九州地区（福岡）	福岡・佐賀・長崎・熊本・大分 宮崎・鹿児島・沖縄	〒816 福岡市博多区井相田2-12-1 シャープエレクトロニクス販売㈱ 九州統轄（営） パソコン営業部、北山部長・岩崎課長 ☎092-501-6806
12月	首都圏地区（東京）	埼玉・山梨・千葉・新潟・東京	〒162 東京都新宿区市ヶ谷八幡町8 シャープエレクトロニクス販売㈱ 首都圏統轄（営） パソコン営業部、福井部長・前田課長 ☎03-3266-8248
12月	近畿地区（大阪）	滋賀・京都・大阪・兵庫・奈良・和歌山	〒556 大阪市浪速区恵美須西1-2-9 シャープエレクトロニクス販売㈱ 近畿統轄（営） パソコン担当、岡本課長・細川係長 ☎06-631-1181㈹
2月	補戦（大阪）	全国	〒545 大阪市阿倍野区長池町22-22 シャープ㈱電子機器事業本部システム機器営業部 ☎06-621-1221 ㈹

68買ったらEXEクラブに入ろう！

本体同梱の入会申込ハガキを送るだけで、無料入会。3つのメリット！
メリット1：会員No.入りオリジナル**会員証電卓**がもらえる。
メリット2：各種フェアご優待・イベントご案内等、数々の特典あり。
メリット3：X68000の活用情報が手に入る「**EXEおみこし活動**」に参加できる。
※「申込ハガキををなくしてしまった」という方は、右記「おみこし活動隊」までお電話ください。

EXEおみこし活動とは？
コミュニケーションペーパー「おみこしPRESS」を通じて会員同士が情報を交換、どこまでもX68000を使いこなして盛り上がってしまおう！というのが、その目的。68へのラブコール、会員独自のテクニック・活用法など、あなたの68自慢を「おみこし活動隊」までどうぞ。会員メッセージは随時「おみこしPRESS」に掲載します。

さらに熱心な会員のために、「**おみこしかつぎ人**」制度も設けました。「かつぎ人」3つのメリットは…❶X68000情報交換会「**おみこしかつぎ人の集い**」に参加できる。❷68最新ソフト・各周辺機器が一覧できる「**ソフトウェア・フィールド**」を半年1回送付。❸「**おみこしPRESS**」毎号送付。「かつぎ人」になれば68ユーザーとして一層充実すること間違いなしです。

●「おみこしかつぎ人」になるには、年会費（おみこしかつぎ代）が必要です。個人入会3,000円／グループ入会（5人1組）2,500円・郵便振込にて申込受付。●詳細は店頭の「おみこしPRESS」をご覧になるか、または「おみこし活動隊」にお電話ください。
おみこし活動隊☎（06）886-0354

●お問い合わせは…シャープ㈱電子機器事業本部システム機器営業部 〒545 大阪市阿倍野区長池町22番22号 ☎（06）621-1221（大代表） シャープ株式会社

C-TRACE

CGコンペティション'91　発表！

C-TRACE CGコンペティション'91に多数のご応募ありがとうございました。
審査の結果、グランプリは、牛澤敏彦様に決定しました。

グランプリ

「Happy Birthday」
牛澤敏彦　様

作者のコメント：
私の愛娘のお誕生日にブタ君とタヌキ君がケーキとプレゼントを持ってやってきました。娘はまだ1才9ヶ月ですが、3才のバースディを想定しています。C-TRACEによる光と影のファンタジーを目指しています。

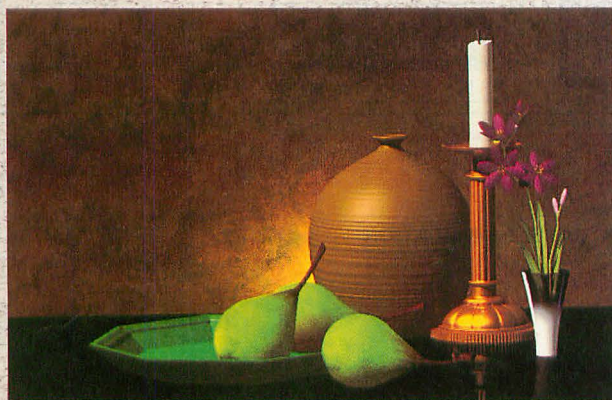
準グランプリ

「アトリエ#1　洋梨のある静物」
稲見　薫 様

「遠方見聞録II」
熊野　務 様

「くもり」
下田達也 様

「ETO(Electric Tripping Object)」
矢野　良 様

「カエル」
河本　保 様

金賞	「Pencil」 小川択延 様	銅賞	「DINOSAURS」 荒井　清 様
銀賞	「Welcome to Ryugu」 神谷淑貴 様	ステゴちゃん賞	桐谷佳典 様、畠山　尚 様、 林　秀則 様

アートディンクフェア

SIMULATION GAME
RAILROAD MANAGEMENT
URBAN DEVELOPMENT

A.III.
A列車で行こう

※画面写真は全て98版のものです。

レールを敷き、駅をつくる。19種類の列車を走らせる。
もちろん、効率的なダイヤを組んでだ。土地を買い、資材を運び、
工場を建てる。ビル、デパート、スタジアム……
四季の移り変わりとともに、街はやがて都市となる。
「AIII」がついにX68000の世界へやってきた……!

X68000……**¥9,800**(税込)

AIII MAP CONSTRUCTION
「AIIIマップコンストラクション」 **¥3,000**(税込)
「AIII マップコンストラクション・新マップ付」 **¥4,000**(税込)

X68000 好評発売中!

¥7,000(税込)

¥7,000(税込)

¥7,000(税込)

- ●A列車で行こうII 新マップ付……**¥8,000**
- A列車で行こうII 新マップのみ……**¥2,500**
- ●大海令……**¥8,000**
- 大海令 シナリオDE……**¥2,500**
- 大海令 シナリオFG……**¥2,500**
- ●ダブルイーグル……**¥7,000**
- ダブルイーグルトリッキーホール……**¥2,000**
- ●南海の死闘……**¥6,000**
- 南海の死闘 シナリオ……**¥2,500**

※価格はすべて消費税込みの価格です。

brother
パソコンソフト自販機

札幌	そうご電器YES 5F	(011)214-2850
	パソコンショップハドソン	(011)205-1590
青森	電巧堂チェーン青森本店	(0177)23-2356
盛岡	デンコードー盛岡本店	(0196)54-2772
秋田	電巧堂チェーン秋田駅前本店	(0188)34-3151
仙台	デンコードー仙台本店	(022)225-5290
	デンコードーDac仙台東口店	(022)291-4744
	庄子デンキコンピュータ中央店	(022)224-5591
郡山	うすい百貨店	(0249)32-0001
福島	庄子デンキ新又駅前店	(0245)21-2011
いわき	いわきマイコンショップ	(0246)23-0513
山形	庄子デンキ山形七日町本店	(0236)42-1222
新潟	PIC	(025)243-5135
	PICこばり店	(025)233-5791
長岡	丸専デパート 1F	(0258)33-4970
	丸専マイコンショップジャスコ長岡店	(0258)27-6033
宇都宮	K&P宇都宮	(0286)62-0002
前橋	パソコンランド21 前橋店	(0272)21-2721
高崎	パソコンランド21 高崎店	(0273)26-5221
太田	パソコンランド21 太田店	(0276)45-0721
桐生	パソコンランド21 桐生店	(0277)45-2721
水戸	川又書店駅前店 2F	(0292)31-0102
つくば	ミドリ電気館	(0298)55-3715
大宮	ダイエー大宮店 6F	(0486)45-4147
川越	長崎屋川越新宿電気館 1F	(0492)44-5461
上尾	ボンベルタ上尾	(0487)73-8711
所沢	サン家電所沢店 2F	(0429)39-5117
松戸	イトーヨーカドー松戸店	(0473)68-5131
千葉	ラオックス千葉店(PERIE 4F)	(0472)27-5318
八千代台	ラオックス八千代台店	(0474)85-2261
秋葉原	丸善無線ECCS 4F	(03)3255-4911
	ミナミ電気館 4F	(03)3255-4040
	サトームセン(メディアセンター)	(03)5256-3267
	サトームセン(ラジオ館) 5F	(03)3251-1464
	サトームセン本店 5F	(03)3253-5879
	CVA秋葉原 2F	(03)3258-3711
	コム本店	(03)3251-1523
	ザ・コンピュータ館	(03)5256-3111
新宿	ラオックス新宿店	(03)3350-1241
池袋	ビックカメラ池袋東口本店 4F	(03)3988-0020
	ビックカメラ池袋北口店 4F	(03)3988-6666
渋谷	上新電機J&P渋谷店 1F	(03)3496-4141
	ビックカメラ渋谷店A館 6F	(03)3477-0002
武蔵野	ラオックス吉祥寺店 2F	(0422)21-3471
小金井	サン家電小金井店	(0423)85-3810
立川	丸善無線立川店(Will 6F)	(0425)27-6211
八王子	ムラウチ電気 2F	(0426)42-6211
	J&P八王子店(そごう 7F)	(0426)26-4141
町田	東急ハンズ町田店 B1F	(0427)28-2603
	J&P町田店	(0427)23-1313
横浜	ソフトクリエイト横浜店	(045)314-4777
	横浜VIVRE21 7F	(045)314-2121
	ダイエー戸塚店 3F	(045)881-1261
	ICコスモランドあざみの	(045)901-1901
	ICコスモランドかもい	(045)934-9636
川崎	セキグチデンキ館	(044)244-5421
藤沢	WAVE EYE	(0466)43-1771
平塚	梅屋	(0463)22-4147
厚木	ラオックス厚木店オーディオ館	(0462)22-2722
大和	WAVE EYE大和店	(0462)63-9898
甲府	中込電気商会	(0552)24-5431
長野	ダイエー長野店 7F	(0262)27-1311
松本	遠兵	(0263)32-6350
金沢	うつのみや片町店	(0762)21-6136
富山	丸の内カラー駅前店	(0764)41-9075
福井	エジソン	(0776)26-2228
	だるまや西武 7F AVC	(0776)27-0111
沼津	メルバ沼津	(0559)22-4858
静岡	メルバ静岡	(0542)54-5338
富士	メルバ富士店	(0545)51-8022
浜松	メルバ浜松本店	(0534)64-8412
	ホーエー家電有楽店	(0534)53-1441
	マエダM&V	(0534)73-1691
名古屋	栄電社テクノ名古屋	(052)581-1241
	パソコンショップコムロード	(052)263-5828
	ちくさ正文館書店ターミナル店	(052)732-3601
	カトー無線電機電気館 4F	(052)264-1534
	丸善無線 第一アメ横店	(052)263-1626
	栄電社テクノ大須	(052)252-0181
	トップカメラ 4F	(052)971-0111
	J&P大須店	(052)262-1141
	エイデンメディア大須	(052)242-8591
豊橋	栄電社テクノ豊橋	(0532)52-1231
岡崎	ジャスコ岡崎店 3F	(0564)23-4960
豊田	ジャスコ豊田店パソコンショップJPC	(0565)35-1761
岐阜	パソコンショップコムロード岐阜店	(0582)66-0288
大垣	スイテック大垣ヤナゲン店A館 6F	(0584)81-3491
津	Kawai無線OA 津店	(0592)26-0111
四日市	Kawai無線カテンワールド四日市	(0593)54-3366
松坂	Kawai Bax店 6F	(0598)26-0111
大津	西武百貨店関西大津	(0775)25-0111
日本橋	J&Pテクノランド	(06) 634-1211
	J&Pメディアランド	(06) 634-1511
	ニノミヤエレランド	(06) 632-2038
	ニノックスコア日本橋店	(06) 647-2038
難波	ニノミヤパソコンランド難波店	(06) 643-3217
梅田	ニノミヤパソコンランド駅前第4ビル店	(06) 341-2031
	J&P阪急三番街店	(06) 374-3311
高槻	上新電機J&P高槻店	(0726)85-1212
八尾	西武百貨店関西八尾店	(0729)97-0111
京都	J&P京都寺町店	(075)341-3571
和歌山	J&P和歌山店	(0734)28-1441
神戸	星電社三宮本店C・ソフト	(078)391-8171
	ダイエー三宮電器館パレックス	(078)391-7911
	上新電機三宮一番館	(078)211-2111
姫路	上新電機J&P姫路店	(0792)22-1221
	姫路コンピュータソフトコンパックス	(0792)94-8244
岡山	岡山VIVRE21	(0862)32-8881
倉敷	倉敷ファソランド	(0864)25-4701
広島	松本無線パーツ	(082)243-4451
	ダイイチパソコンCITY	(082)248-4343
高松	Sound Check MOVE	(0878)61-6171
徳島	徳島そごう 7F	(0886)25-4056
高知	デンキのタグチ	(0888)23-0181
松山	ダイイチ松山パソコンCITY	(0899)31-6711
北九州	ベストマイコン小倉パソコン館	(093)551-6281
福岡	ベストマイコン福岡店	(092)781-7131
	寿屋エレデ博多	(092)281-4411
長崎	ベスト電器長崎新地店	(0958)28-1333
熊本	寿屋本荘店	(096)372-5411
	ベストマイコン熊本パソコン館	(096)332-4180
鹿児島	ベスト電器鹿児島パソコン館	(0992)23-2081
大分	ベスト電器大分パソコン館	(0975)32-9396
宮崎	宮崎寿屋百貨店 7F	(0985)27-4111
那覇	ベスト電器那覇店 6F	(0988)62-7588

もう、どこかで見ましたか?「NEW TAKERU」。「ソフトベンダー武尊」の生まれ変わった新しい姿を。どこが変わったと思いますか?色?そう、色も変わっていますね。でも、ホントはもっとスゴイ所が変わったんです!いってみれば、TAKERUの「脳みそ」がイッキにかしこくなったって感じかな…?だから通信時間もソフトの書き込み時間もマニュアルの印字時間も、みんなみんなスピードアップ!だからもう、お待たせしません!!他にも1万円札が使えたり、レーザープリンターが搭載されたり、TAKERU CLUBができるetc…たくさんあって伝えきれないっっ!!夏頃には、キミの近くの「武尊」も「NEW TAKERU」に。楽しみに待っててネ!

ブラザー工業株式会社
〒467 名古屋市瑞穂区苗代町2番1号
新事業推進室
TAKERU事務局
(052)824-2493
東京営業所(03)3274-9616
大阪営業所(06)252-4234

通信販売
通信販売をご希望の方、ソフト名・機種名・住所・氏名・電話番号を明記の上、TAKERU事務局まで現金書留でお申し込み下さい。
代金引換えも始めました。詳しくは、TAKERU事務局通信販売係まで!!

ゴルフゲームのスタンダード！

LICENSED BY
AUGUSTA NATIONAL GOLF CLUB

マスターズの興奮が今、蘇る。

neXt
RPG・ACT・SLG、最強のラインナップで
次世代体験……neXt！

オーガスタ・ナショナル
ゴルフ・クラブと正式契約

X68000版 好評発売中!!

×68000版特長

- 実際にゴルフコースに立った状態と同じ視野でプレイ可能。
- どの地点にきても全方向の視野画面をリアルタイム3D表示。
- ホールすべてにアンジュレーション（起伏）を3Dで表示。
- ボールの落下地点の状態によってバウンド、転がり等が本物同様に変化。
- ストロボアクションモードでボールの軌跡を確認可能。
- トップスピン・バックスピンも自在、キャディーは4人の中から選択。
- ADPCMによるリアルなサウンド。ショット音、歓声、小鳥のさえずりまでも忠実に再現。
- プレイモードは3種類。ストロークプレイ、マッチプレイ、トーナメントプレイ。
- スコア・各種個人データ等を自動保存、プリントアウトも可能。
- 初心者でも手軽に楽しめるスローモード機能あり。
- 31KHz、15KHz両モード対応。

POLYSYS™
Integrated 3D Processor

このマークはT&E SOFTの商標です。
POLYSYS搭載の3Dソフトには、このマークが表示されます。

RPG-neXt……ルーンワース 黒衣の貴公子
ACT-neXt……幻 獣 鬼
SLG-neXt……遥かなるオーガスタ

X68000（5"2HD 3枚組）要2M RAM

- PC-9801VMシリーズ（5"2HD 2ドライブ）要640K RAM
- PC-9801UVシリーズ（3.5"2HD 2ドライブ）要640K RAM
- PC-9801N/URシリーズ（NOTE専用版）（3.5"2HD 1ドライブ+1RAMドライブ）

※上記のソフトはエプソンPC-286,386シリーズに対応

- FM TOWNS（CD-ROM & 3.5"2HD）要2M RAM

標準価格 各¥12,800（税別）

Technology & Entertainment Software

株式会社 ティーアンドイーソフト

No Copy
このマークは不法コピー禁止マークです
ソフトウエア法的保護監視機構

〒465 名古屋市名東区豊が丘1810番地 PHONE:052-773-7770

- 3Dゴルフに関するお問い合わせは、NEW 3D GOLF事務局まで PHONE:052-773-7757

2億年もの太古の世界に繰り広げられる想像を越えた物語。

マーキュリー ザ・プライム・マスター

MERCURY

THE PRIME MASTER

マーキュリー ザ・プライム・マスター

X68000 ¥8,800(税別) 好評発売中

- ◆アイソメトリックビュー(等角投影画面)により立体感あふれる3Dフィールドを実現!
- ◆マウス操作による快適なゲーム進行を実現。アイコンやオブジェクトをクリックするだけでアイテム使用や戦闘が可能。
- ◆魔法と科学が共存する世界で繰り広げられる数々のドラマ!
- ◆全23曲の美しいBGMが物語の興奮をさらに盛り上げ、今までに存在しなかったRPG世界を奏でる。

MAXIMA

■制作・発売元

マキシマ

大阪市浪速区塩草3-3-26 TEL(06)561-2215

札幌 TEL(011)612-0610・広島 TEL(082)924-5061

●お求めはお近くのパソコンショップで、通信販売をご希望の場合は使用機種名、住所、氏名、電話番号を明記の上、当社まで現金書留でお申し込みください。(送料無料)

未来とは定められた運命なのか？

人類の歴史は偶然の結果の記録ではない。
それは、定められたひとつの目的にしたがって操作された結果である。
あらゆる予言の書の存在…。
なぜ人が未来を知り得るのであろうか？
人類は定められた運命を変える事ができるのか？

初回出荷限定版
オリジナル・マウスマット、ライセンスカード付

スペキュレーティブ・アドベンチャー

シグナトリー

SIGNATORY

――調印者――

提供■NCS　制作総指揮・総監督・原作■鈴木 力　脚本■成田伸子　出演■ケニー・フィリップ/バーバラ・ドゥーティ/トーマス・スウェイジ他　制作■Tenky

■制作スタッフ■スクリプト/皆川正三 ■SE・プログラム/橋谷利幸 ■メイン・プログラム/Hかずき ■チーフデザイン/石井秀明 ■デザイン/本間繁二郎/大村政幸/矢田 智/古澤雅子 ■音楽・効果音/高橋大昌 ■NY・南米取材/青空風太郎 ■NY取材協力/氷上 情

■マウスオペレーションで簡単操作 ■200枚を超える美しいグラフィック
■史実の謎に迫る野心的ストーリー ■現地取材をもとにしたリアルな構成

X68000 ONLY 5'2HD(5枚組) 価格¥12,000(税抜)

「シグナトリー」の世界に迫る

総監督鈴木力氏自ら「シグナトリー」を語る。

「シグナトリー」は今までのパソコンゲームにはないテーマ設定がなされた意欲作である。

人類の辿ってきた軌跡と未来についての様々な諸説、それらを基にフィクション化し、ゲーム化している。いや、単なるゲームでは語り切れない世界観が感じられる。それを鈴木氏に語ってもらった。

「何故、人が未来を知る事ができるのか？」ミッシェル・ノストラダムスという人は彼の「予言集」に未来を書き記した。そして、それは現実に的中している。

先の問いを発端に、私は一つの疑問を考え続けてきた。未来とは確定した事実なのだろうか？

これに対する一つの回答となる仮説は、全く別の事柄に目を向けた時に、わたしの頭の中に浮び上がった。

それがアドルフ・ヒトラーであり、月に関する謎である。我々が学校教育で習った「歴史」とは、単なる「年表」であって「歴史」そのものではない事を痛感する。

第二次世界大戦が何を生み出し、それが後世に何を残したのか。少なくともナチス・ドイツに限っては「悪魔」のイメージだけであって、当時最も先端を進んでいたドイツの科学技術には何ら触れていない。

例えば戦後、アメリカ軍がドイツから押収した科学開発関係の文献や図面等を解読するために、わざわざ航空専門用語の独英辞典が作られたほどである。あるいはまた、アメリカに渡ったドイツの科学者たちの素晴らしい業績は、「アメリカ航空機年鑑」にも記されている。

ドイツの科学技術者はアメリカだけでなく、ソビエトにも渡っている。戦後のアメリカとソビエトは、ドイツの技術者を自国に取り込む事で大きく発展したといっても過言ではない。ナチス・ドイツは我々に遺産を残した。これは、否定できない事実なのだ。(註：私はナチスを崇拝してはいない。これは事実なのだ)

もう一方の敗戦国である日本については、どうであろうか。戦後、戦勝国による分割統治案もあったが、我が国は奇跡的にその運命を免れている。

ドイツは分割され、半世紀近く東西に分けられるという非運にあっているというのに、日本は占領される事なく復興できたばかりかやがて先進国家へ変貌する。経済ではアメリカをも脅かす存在になってしまう。これは常識ではあまりに不自然であり、異様としか思えない。

第二次世界大戦後の世界構造が、どれだけ異質なものであるかを考えてほしい。私はこの異質さに、何か自然発生的ではない作為的なものを感じざるを得ない。

「そうなるように、あらかじめ決定されていたのでは？」

そう確信を持てるようになったのは「月」についての資料を読んだ後である。ご存知のように月は地球の生命に多大な影響を与えている。その月の存在が、今もって科学的に説明できないという事実がある。「月」とは矛盾を抱えた衛星なのである。

1969年のアポロ計画により、月面写真は総計十万枚以上になる。しかしその大部分は「極秘資料扱い」として一般に公開されていない。天体の観測写真のほとんどが、国家保安上の名目で極秘扱いとなる事自体が異常である。一般の人に見せられないほどの重大機密が「月」にあるのだろうか？

どうやら、それがあるらしい。興味のある方はぜひ、『月は神々の前哨基地だった(たま

出版)』を読んで欲しい。その中の月面写真のすべてではないにしても、どう見ても自然現象とは思えないものがある。そして正式発表された写真には修正が行なわれているという事実が、これを決定的にしている。

自然の天体と思っていた「月」が人工物であるらしい。だとすれば、地球の生命もまた人工的に手を加えられているのでは？そしてそれは我々の歴史にも及んでいるのでは？誰が？何のために？

我々は知るべき事を知らされず、(人間以外の存在である事は間違いない)何者かの手に未来を握られてしまっているのではないのか？全てはこの疑問からはじまった。その意味ではこのゲームは異質である。なぜなら、全ては「完全なる空想」ではない。これは「可能性として考えられる仮説」をもとにした空想なのである。

独特の世界観で創られた「シグナトリー」。商品に付属しているプレミアムブックで、この世界の設定を読み、実際にプレイして何かを感じて欲しい、そう鈴木氏は語る。

無論ゲームとして楽しめるよう、007ばりの迫真のアドベンチャーに仕上げてある。

■店頭にて、新作ゲームソフト25~30%OFF!!（税別）、超低金利オクトハッピークレジットをご利用下さい!!

■朗報デス。夏のボーナス一括（7月末）払いOK!!手数料無料。ご利用下さい。■店頭にて、新作ゲームソフト25~30%OFF!!

■便利です。夜9時まで営業しております。お立寄り下さい。お待ちしております!!

パソコンプラザ OCT（オクト）

案内図

JR蒲田から徒歩5分
京浜蒲田から徒歩1分

2F 商店街アーケード

店頭セール実施中

オクトで始まるパソコンワールド

☎03-3730-6271

●営業時間 AM 11:00～9:00/日曜・祭日PM7:00　電話一本で、ハイ即納

〒144 東京都大田区蒲田4-6-7 FAX 03-3730-6273

全国通販

●定休日毎週火曜日 祭日の場合翌日になります。

オクト ラクラククレジット

3回	3.5%	6回	4.5%	10回	6%	12回	6%	18回	11%
20回	12%	24回	12.5%	36回	17.5%	48回	23%	60回	29.5%

OCT-1 システム インフォメーション

- ▶全商品保証付（メーカー保証）
- ▶超低金利ハッピークレジット（1回～60回）頭金ナシOK!
- ▶ボーナス一括払いOK! ボーナス2回払いOK!!
- ▶配達日の指定OK!（万全なサポート体制）
- ▶商品の組合せ自由! オクトフリーダムシステム
- ▶店頭デモンストレーション実施中

オクト セレクテッドシステム

広告掲載商品以外の製品も取扱っております。

蒲田

夏のボーナス一括（7月末）払いOK!!
X68000XVIデビュー記念セール実施中!!
あなたもTELしてこの感動を…!! NOW ON SALE

X68000 XVI エクシヴィ

■16MHz■
■SX-WINDOW ver1.1■
■Attachment MEMORY BORD■

快速16MHz 鮮烈デビュー!!

■CZ-634C-TN

定価￥368,000

Ⓐ●CZ-634C-TN
●CZ-613D-TN
定価合計￥503,000

12回	?	24回	?	36回	?	48回	?	60回	?

Ⓑ●CZ-634C-TN
●CZ-606D-TN
定価合計￥447,800

12回	?	24回	?	36回	?	48回	?	60回	?

■CZ-644C-TN

定価￥518,000

Ⓒ●CZ-644C-TN
●CZ-613D-TN
定価合計￥653,000

12回	?	24回	?	36回	?	48回	?	60回	?

Ⓓ●CZ-644C-TN
●CZ-606D-TN
定価合計￥597,800

12回	?	24回	?	36回	?	48回	?	60回	?

X68000XVI
新発売記念プレゼント
あなたのオクトから素適な贈物!!

今、XVIをお買い上げいただいたあなたに①又は②と③番の商品をプレゼントしちゃいます!!

① 遥かなるオーガスタ 大人気 大戦略II（キャンペーン版）

ゴルフゲームの決定版　不朽の名作X68000版

（定価￥12,800） + （定価￥9,800）

or

② インテリジェントコントローラ
■CZ-8NJ2（CYBER STICK）
シューティングゲーマーの必須アイテム!!

（定価￥23,800）

③ MD-2HD（10枚）
シリコンキーボードカバー
もれなく!! サービス!!

▶現金超特価 ￥TEL下さい!!▶

※どちらかお選び下さい!!（どっちが得かヨーク考えてネ!）

特選周辺機器（送料￥500）

- ●SX-68M MIDIインターフェースボード（システムサコム）￥19,800…特価￥14,500
- ●Fine Scanner X68（HAL研究所）（HGS-68）￥39,800…特価￥26,300

■増設RAMボード＝I・Oデータ

① PIO-6BE1-A（1MB）￥25,000…特価￥17,000
② PIO-6BE2-2M（2MB）￥50,000…特価￥34,800
③ PIO-6BE4-4M（4MB）￥88,000…特価￥60,000

周辺機器コーナー（送料￥500）

商品	定価	特価
●CZ-6BEI IBM増設RAMボード	（￥35,000）	▶特価￥26,000
●CZ-6BEIB IBM増設RAMボード	（￥28,000）	▶特価￥21,000
●CZ-6BE2 2MB増設RAMボード	（￥79,800）	▶特価￥60,000
●CZ-6BE4 4MB増設RAMボード	（￥138,000）	▶特価￥103,000
●CZ-6BFI 増設用RS-232Cボード	（￥49,800）	▶特価￥38,000
●CZ-6BGI GP-IBボード	（￥59,800）	▶特価￥48,000
●CZ-6BMI MDIボード	（￥26,800）	▶特価￥20,200
●CZ-6BNI スキャナ用パラレルボード	（￥29,800）	▶特価￥22,500
●CZ-6BPI 数値演算プロセッサボード	（￥79,800）	▶特価￥60,000
●CZ-6BOI ユニバーサルI/Oボード	（￥39,800）	▶特価￥30,500
●CZ-6EBI/BK 拡張I/Oボックス	（￥88,000）	▶特価￥65,800
●CZ-6VTI/BK カラーイメージ・ユニット	（￥69,800）	▶特価￥52,300
●CZ-8NM2A マウス	（￥6,800）	▶特価￥5,300
●CZ-8NTI マウストラックボール	（￥9,800）	▶特価￥7,500
●CZ-8NSI カラーイメージスキャナ	（￥188,000）	▶特価￥137,000
●CZ-6BCI FAXボード	（￥79,800）	▶特価￥60,500
●CZ-8TM2 モデムユニット	（￥49,800）	▶特価￥38,000
●CZ-64H 増設ハードディスク	（￥120,000）	▶大特価
●CZ-6TU GY/BK RGBシステムチューナー	（￥33,100）	▶特価￥25,000
●BF-68PRO 高性能CRTフィルター	（￥19,800）	▶特価￥15,500
●CZ-6MOI 光磁気ディスクユニット	（￥450,000）	▶特価￥328,000
●CZ-6BSI SCSIインターフェースボード	（￥29,800）	▶特価￥22,200
●CZ-6BL2 LANボード	（￥298,800）	▶特価￥220,000
●CZ-6BVI（ビデオボード）	（￥21,000）	▶特価￥15,500
●CZ-6BE2A 2MB増設RAMボード	（￥59,800）	▶特価￥44,500
●CZ-6BE2B 2MB増設メモリ（チップ型）	（￥54,800）	▶特価￥41,000
●CZ-6BP2 数値演算プロセッサ	（￥45,800）	▶特価￥34,000

※クレジットの回数は1回～60回、ボーナス併用などありますのでお電話でお問合せ下さい。

■本体セット：送料無料（注）本体セット以外の周辺機器（プリンター、モデム、HDD等）及びソフトの送料は、北海道・九州地区＝1ケロ￥1500、■その他離島地区は、1ケロ￥2000となります。

※上記料金には、消費税は含まれておりません。消費税が付加されますので、詳しくは、電話でお問合せ下さい。

■特に人気のある商品によっては、しばらくお待ち願うことがありますのでご了承下さい!!

X68000

■ SUPER／PROII／SUPER-HD

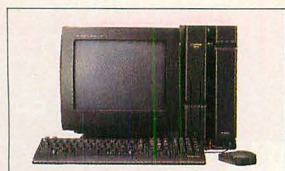
ラストチャンス!!

◀大人気!!*大戦略II シュミレーションゲーム プレゼント

★JOY CARD (連射式)×2個
★MD-2HD 10枚

(定価¥9,800)

■超低金利クレジットをご利用下さい。1回〜60回払い、頭金ナシ!!ボーナス1回及び2回払いOKです。

■ビッグバーゲンセール実施中!!ゲームソフト(ビジネス)新製品続々入荷中!!

■SUPER(定価¥348,000)
CZ-604C-TN

■PRO II (定価¥285,000)
CZ-653C-BK/GY

■SUPER-HD(定価¥498,000)
CZ-623C-TN

15型カラーディスプレイTV

CZ-613D-GY/BK
定価¥135,000

14型カラーディスプレー

CZ-606D(GY/BK/TN)
定価¥79,800

21型カラーディスプレイ

CU-21HD
定価¥148,000

CZ-8NJ2 限定
●インテリジェントコントローラ
定価¥23,800
超特価¥18,000

Ⓐ CZ-604C+CZ-613D…定価合計¥483,000▶オクト大特価

12回	¥30,000	24回	¥15,900	36回	¥11,000	48回	¥ 8,700	60回	¥7,300

Ⓑ CZ-653C+CZ-613D…定価合計¥420,000▶オクト大特価

12回	¥25,600	24回	¥13,600	36回	¥ 9,400	48回	¥ 7,400	60回	¥6,200

Ⓒ CZ-623C+CZ-613D…定価合計¥633,000▶オクト大特価

12回	¥39,300	24回	¥20,900	36回	¥14,500	48回	¥11,400	60回	¥9,600

Ⓓ CZ-604C+CZ-606D…定価合計¥427,800▶オクト大特価

12回	¥26,300	24回	¥14,000	36回	¥ 9,700	48回	¥ 7,600	60回	¥6,400

Ⓔ CZ-653C+CZ-606D…定価合計¥364,800▶オクト大特価

12回	¥22,200	24回	¥11,800	36回	¥ 8,200	48回	¥ 6,400	60回	¥5,400

Ⓕ CZ-623C+CZ-606D…定価合計¥577,800▶オクト大特価

12回	¥35,800	24回	¥19,000	36回	¥13,200	48回	¥10,300	60回	¥8,700

Ⓖ CZ-604C+CU-21HD‥定価合計¥496,000▶オクト大特価

12回	¥31,100	24回	¥16,500	36回	¥11,400	48回	¥ 9,000	60回	¥7,600

Ⓗ CZ-653C+CU-21HD‥定価合計¥433,000▶オクト大特価

12回	¥26,800	24回	¥14,200	36回	¥ 9,900	48回	¥ 7,700	60回	¥6,500

Ⓘ CZ-623C+CU-21HD‥定価合計¥646,000▶オクト大特価

12回	¥40,700	24回	¥21,600	36回	¥15,000	48回	¥11,800	60回	¥9,900

★本体セットは、1ヶ月間だけの大特価セール!!
★クレジット価格は、消費税込みですョ。ご利用下さい!!

X68000ソフト大セール実施中!! (ゲームソフト25〜30%OFF) 送料¥500

〈グラフィック〉●Z's STAFF PRO68K Ver.2.0 (シャフト)定価¥58,000 ……特価¥39,400

〈開発ツール〉●C-コンパラPRO68KV.2 定価¥44,800 CZ-245IS ……特価¥33,300

〈データベース〉●CARD PRO68K Ver.2.0 CZ-253BS ……大特価

〈グラフィック〉●C-TRACE+ 定価¥198,000 ……特価¥145,000

〈C言語〉●C&Professional Pack 定価¥58,000 ……特価¥41,000

〈音楽〉●Music studio PRO68K Ver.2.0 定価¥28,800 CZ-261MS ……特価¥21,500

〈CGツール〉●CANVAS PRO68K 定価¥29,800 CZ-249GS ……特価¥22,200

〈ワープロ〉●Multiword PRO68K CZ-225BS ……大特価

〈通信〉●Tlepotion PRO68K CZ-258BS ……大特価

熱転写カラー漢字プリンター (ケーブル用紙付) 送料¥1,000

■CZ-8PC5 NEW

●48ドット
●熱転写カラー漢字プリンター
定価¥96,800
特価¥69,800

①CZ-8PK10(24ピン漢字プリンター136桁) 定価¥97,800……大特価!!TEL下さい。
②CZ-8PGI(24ピンカラー漢字プリンター80桁) 定価¥130,000……大特価!!TEL下さい。
③CZ-8PG2(24ピンカラー漢字プリンター136桁) 定価¥160,000……大特価!!TEL下さい。
③IO-735×(カラーイメージジェット) 定価¥248,000……大特価¥177,000

モデルコーナー 送料¥1,000

オムロン
●MD-1200AIII……特価¥14,500
●MD-12FS……特価¥15,000
●MD-24FP4II……特価¥26,000
●MD-24FN5……特価¥30,000
●MD-24FS4……特価¥31,000
●MD-24FS5……特価¥30,000
●MD-24FS7……特価¥43,500
●MD-24FC5……特価¥34,000
●MD-24FP5II……特価¥28,500
●MD-24FN4……特価¥27,000
●MD-24FJ4……特価¥31,000
●MD-24FJ5……特価¥34,000
●MD-24HS……特価¥64,000
●MD-48HS……特価¥98,000
●MD-96FS5……特価¥131,000

AIWA
●PV-A24VM5……特価¥29,000
●PV-M24VB5……特価¥30,000
●PV-A12……特価¥14,500
●PV-M24……特価¥28,500

X68000 SOFT WARE 送料¥500〜

型名	商品	定価	特価
CZ-212BS	BUSINESS PRO-68K	¥ 68,000	¥ 48,000
CZ-213MS	MUSIC PRO68K	¥ 18,800	¥ 13,500
CZ-214MS	SOUND PRO-68K	¥ 15,800	¥ 11,500
CZ-215MS	Sampling PRO-68K	¥ 17,800	¥ 12,800
CZ-219SS	OS-9/X68000	¥ 29,800	¥ 21,000
CZ-220BS	DATA PRO-68K	¥ 58,000	¥ 41,000
CZ-223CS	Communication PRO-68K	¥ 19,800	¥ 14,300
CZ-224LS	THE 福袋 V2.0	¥ 9,900	¥ 7,500
CZ-241BS	システム手帳リフィル集	¥ 9,800	¥ 7,500
CZ-242BS	活用フォーム集	¥ 9,800	¥ 7,500
CZ-244SS	Homan 68K Ver2.0	¥ 9,800	¥ 7,500
CZ-247MS	MUSIC PRO-68K(MIDI)	¥ 28,800	¥ 20,800
CZ-240BS	Stationery PRO-68K	¥ 14,800	¥ 11,500
CZ-243BS	CYBER NOTE PRO-68K	¥ 19,800	¥ 15,200
EW		¥ 38,000	¥ 29,800
G-68K		¥ 14,800	¥ 11,400
E-68		¥ 19,800	¥ 15,300
CZ-255GS	CANVASドローグラフィックLIB	¥ 8,800	¥ 29,600
CZ-256GS	CANVASドローグラフィックVOL.2	¥ 8,800	¥ 6,600
CZ-260LS	XBAS to CHECKER PRO68K	¥ 9,800	¥ 6,600
CZ-259SS	SX-WINDOW Ver.1.0	¥ 6,800	¥ 7,500
CZ-234LS	AI-68K	¥188,000	¥ 5,000
CZ-251BS	ハイパーワード	¥ 39,800	¥139,000
	KAMIKAZE	¥ 68,000	¥ 29,800
	デジタルクラフト	¥ 39,800	¥ 45,400
	サイクロエキスプレスα68	¥ 97,000	¥ 28,000
	C-TRACE 68 Ver3.0	¥ 78,000	¥ 73,000
	C-TRACE TP+	¥398,000	¥ 69,500

パソコンラック 推奨 送料無料

①五段キャスター付

5段キャスター付
キーボードが収納できるから、手元でマウス操作がラクラクできる
棚板5段のマルチに活用できるディスク。
ウーン、こいつはデキル!
1325(H)×640(W)×700(D)
特価¥15,000

②四段キャスター付

4段キャスター付
どんなパソコンにもフレキシブルに対応!
使い易いデスクです。
1245(H)×614(W)×600(D)
特価¥11,000

店頭新作ゲームソフト25〜30%OFF!! ビジネスソフト25%より特価中

★通信販売お申込みのご案内★ 〒144 東京都大田区蒲田4-6-7 TEL:03-3730-6271

お申込みはお電話でお願いします。お客様の〈住所〉〈氏名〉〈電話番号〉及び〈商品名〉をお知らせ下さい。●入金確認後ただちに商品をご送付いたします。

現金一括払い
銀行振込:お近くの銀行より(電信扱い)にてお振込み下さい。
現金書留:封筒の中に住所・氏名・商品名をご記入の上当社までお送り下さい。

クレジット
専用お申込用紙をお送り致します。ので、必要事項をご記入、ご捺印の上ご返送下さい。手続きは簡単です。

オクト ラクラク クレジット表

2回	3.5%	6回	4.5%	10回	6%	12回	6%
15回	9%	18回	11%	20回	12%	24回	12.5%
30回	17.5%	36回	17.5%	48回	23%	60回	29.5%

振込先
富士銀行 久ヶ原(クガハラ)支店 当No.1824
三菱銀行 蒲田支店 当No.0278691
株式会社 億人(オクト)

※掲載の価格は変動しますので、まずは、お電話にてご確認ください。
※上記料金には、消費税は含まれておりません。消費税が付加されますので、詳しくは電話でお問合せ下さい。
※銀行振込、または、現金書留でご注文の際には、あらかじめ電話でご確認の上、お申し込み下さい。

超低金利クレジットOK

注目!! 夏のボーナス一括払い 手数料(金利)無料

(平成3年7月末をご利用下さい)

またまた秋葉原でおなじみの

5/15〜6/14

●お近くの方はお
●本体単品で特
●ビジネスソフト定

HARD DISK UNIT(X68000専用)
アイテック(SCSI) (送料¥1,000)
●ITX-80S(80MB/20ms)·定価¥128,000▶特価¥ 88,000
●ITX-130S(130MB/20ms)…定価¥158,000▶特価¥107,000

Fine Scanner-X68
(HAL研究所)X68000専用
■HGS-68 (定価¥39,800)
特価¥26,300
(送料·消費税込み¥27,604)

X68000シリーズ専用 特価¥14,700
MIDIインターフェースボード
SX-68M(サコム)
(純正コンパチ)定価¥19,800
(送料·消費税込み¥15,759)

X68000メモリボード(シャープ&I/O·DATA)(送料¥500)
①CZ-6BE1(600C用) 定価¥35,000 (送料·消費税込¥27,295)……¥26,000
②PIO-6BE1-A 定価¥25,000 (送料·消費税込¥18,540)……¥17,500
③PIO-6BE2-2M 定価¥50,000 (送料·消費税込¥35,329)……¥33,800
④PIO-6BE4-4M 定価¥88,000 (送料·消費税込¥59,225)……¥59,225

ジョイスティック 送料¥500
●X-1PRO 定価¥9,500▶特価¥7,800
●ASCII STICK 定価¥6,800▶特価¥5,500

X68000-XVI 新発売!! (送料·消費税込み)

NEW

★先着100名様へ。ゲームソフト(V-BALL¥7,900)をプレゼント!!

X68000-XVI▶セットでお買い上げの方に●ディスケット10枚●ジョイカード2ケプレゼント中!!

Ⓐセット:CZ-634C-TN+CZ-606D-TN…定価¥447,800▶特価価格はTEL下さい。

12回	?	24回	?	36回	?	48回	?	60回	?

Ⓑセット:CZ-634C-TN+CZ-613D-TN…定価¥503,000▶特価価格はTEL下さい。

12回	?	24回	?	36回	?	48回	?	60回	?

X68000-XVI-HD▶セットでお買い上げの方に●ディスケット10枚●ジョイカード2ケプレゼント中!!

Ⓐセット:CZ-644C-TN+CZ-606D-TN…定価¥597,800▶特価価格はTEL下さい。

12回	?	24回	?	36回	?	48回	?	60回	?

Ⓑセット:CZ-644C-TN+CZ-613D-TN…定価¥653,000▶特価価格はTEL下さい。

12回	?	24回	?	36回	?	48回	?	60回	?

※上記のモニターを、CZ-604D(定価¥94,800)、CZ-605D(定価¥115,000)、CU-21HD(定価¥148,000)に変更の場合、TEL下さい。超特価で販売致します。

X68000シリーズ〜P&Aスペシャルセット=台数限定 送料、消費税込み

※セットでお買い上げの方に、●ディスケット10枚、●ジョイカード2個プレゼント中!!

先着100名様。ゲームソフト(V-BALL¥7,900)をプレゼント!!

SUPER

Ⓐセット:P&A特選セット
■CZ-604C (本体定価¥348,000)
+
■CZ-606D (モニター定価¥79,800)
▶P&A超特価 ¥306,000

Ⓑセット
■CZ-604C+CZ-604D
定価¥442,800……▶特価¥312,000

Ⓒセット
■CZ-604C+CZ-605D
定価¥463,000……▶特価¥330,000

Ⓓセット
■CZ-604C+CZ-613D
定価¥483,000……▶特価¥345,000

Ⓔセット
■CZ-604C+CU-21HD
定価¥496,000……▶特価¥353,000

PRO-II

Ⓐセット:P&A特選セット
■CZ-653C (本体定価¥285,000)
+
■CZ-606D (モニター定価¥79,800)
▶P&A超特価 ¥242,000

Ⓑセット
■CZ-653C+CZ-604D
定価¥379,800……▶特価¥250,000

Ⓒセット
■CZ-653C+CZ-605D
定価¥400,000……▶特価¥269,000

Ⓓセット
■CZ-653C+CZ-613D
定価¥420,000……▶特価¥283,000

Ⓔセット
■CZ-653C+CU-21HD
定価¥433,000……▶特価¥290,000

SUPER-HD

Ⓐセット:P&A厳選セット
■CZ-623C (本体価格¥498,000)
+
■CZ-606D (モニター定価¥79,800)
▶P&A超特価 ¥382,000

Ⓑセット
■CZ-623C+CZ-604D
定価¥592,800……▶特価¥389,000

Ⓒセット
■CZ-623C+CZ-605D
定価¥613,000……▶特価¥408,000

Ⓓセット
■CZ-623C+CZ-613D
定価¥633,000……▶特価¥420,000

Ⓔセット
■CZ-623C+CU-21HD
定価¥646,000……▶特価¥430,000

EXPERII

Ⓐセット:P&A厳選セット
■CZ-603C (本体価格¥338,000)
+
■CZ-606D (モニター定価¥79,800)
▶P&A超特価 ¥288,000

Ⓑセット
■CZ-603C+CZ-604D
定価¥432,800……▶特価¥294,000

Ⓒセット
■CZ-603C+DZ-605D
定価¥453,000……▶特価¥310,000

Ⓓセット
■CZ-603C+CZ-613D
定価¥473,000……▶特価¥327,000

Ⓔセット
■CZ-603C+CU-21HD
定価¥486,000……▶特価¥329,000

P&A超低金利クレジットをご利用ください!!

●本広告の掲載の商品の価格については、消費税は含まれておりません。●営業時間=平日AM10:00〜PM7:00、日祭AM10:00〜PM6:00

回～84回払いまでOK!!

★頭金なし!★即日発送

●価格は流通事情により変動致しますので、銀行振込・書留等の送付前に、あらかじめお電話にてご確認下さい。

P&Aがズバリ超特価セールでご奉仕!!

立寄り下さい。専門係員が説明いたします。
価で受付します。詳しくは電話にてお問合せ下さい。
価の20%引きOK! TELください。

全国通販

超特価でクレジットが組める!!

X68000用ソフトコーナー(送料1ヶ～5ヶまで￥500)

商品	定価	特価
●Z's STAFF PRO68K Ver.2.0(ツァイト)	￥58,000	￥38,500
●Z's TRIPHONY デジタルクラフト(ツァイト)	￥39,800	￥27,800
●テラッツォ(ハミングバード)	￥19,400	￥14,200
●KAMIKAZE(サムシング・グッド)	￥68,000	￥44,800
●C & Professional Pack(マイクロウェアジャパン)	￥58,000	￥40,500
●Final Ver3.2(エーエスピー)	￥38,000	￥29,600
●C-compiler PRO68K Ver.2 CZ-245L	￥44,800	￥33,300
●CARD PRO68K CZ226BS	￥29,800	￥21,200
●Y.BAS to C CHECKER CZ-260LS	￥9,800	￥7,400
●OS-9/X68000 CZ219SS	￥29,800	￥22,500
●AI-68K CZ234LS	￥188,000	￥138,000
●THE 福袋 V2.0 CZ224LS	￥9,900	￥7,400
●SOUND PRO68K CZ-214MS	￥15,800	￥11,400
●MUSIC PRO68K CZ213MS	￥18,800	￥13,400
●Sampling PRO68K CD215MS	￥17,800	￥12,700
●MUSIC-studio PRO68K CZ-252MS	￥15,800	￥21,400
●MUSIC-PRO68K〔MIDI〕247MS	￥28,800	￥20,700
●New-print Shop 221HS	￥19,800	￥15,500
●Communication 223CS	￥19,800	￥14,200
●Communication Ver.2 CZ-257CS	￥19,800	￥15,500
●C-TRACE68 Ver.3.0(キャスト)	￥98,000	￥69,000
●サイクロンEXPRESSα68	￥98,000	￥69,800
●G68K Ver2 PRO	￥22,000	￥17,500
●SX-WINDOW CZ-259SS	￥6,800	￥4,900
●Gツール(ザインソフト)	￥28,000	￥18,900
●たーみのる2(SPS)	￥17,800	￥13,300
●マジックパレット(ミュージカルプラン)	￥19,800	￥14,500
●Hyper word CZ-251BS	￥39,800	￥29,600

●ゲームソフト20%OFF OK!! (一部ソフト除く)

周辺機器コーナー(送料￥500)

商品	定価	特価
①CZ-8NSI	￥188,000	￥145,000
②CZ-6VTI	￥69,800	￥52,500
③CZ-6TU	￥33,100	￥24,500
④BF-68PRO	￥19,800	￥15,300
⑤CZ-6BEI	￥35,000	￥26,000
⑥CZ-6BEIA	￥38,000	￥28,600
⑦CZ-6BE2	￥79,800	￥60,000
⑧CZ-6BE4	￥138,000	￥103,000
⑨CZ-6BFI	￥49,800	￥38,200
⑩CZ-6BPI	￥79,800	￥60,000
⑪CZ-6BMI	￥26,800	￥20,300
⑫CZ-6EBI	￥88,000	￥66,500
⑬AN-S100	￥36,600	￥28,500
⑭CZ-6SDI	￥44,800	￥35,000
⑮CZ-6BN1	￥29,800	￥22,600
⑯CZ-6BV1	￥21,000	￥15,900
⑰CZ-64H	￥120,000	￥91,500
⑱CZ-6BG1	￥59,800	￥45,000
⑲CZ-6BU1	￥39,800	￥30,300
⑳CZ-6PVI	￥198,000	￥153,000
㉑CZ-6BS1	￥29,800	￥22,300
㉒CZ-8NJ2	￥23,800	￥18,500
㉓CZ-6BL2	￥298,000	￥222,000
㉔JX-100S	￥89,800	￥38,800
㉕JX-220	￥146,000	￥107,900
㉕IO-735X	￥248,000	￥169,000

X68000用ハードディスク(送料￥1,000)

アイテム
- ●HXD-040(40MB/23ms)…定価￥118,000▶特価￥ 88,000
- ●HXD-042(増設用)…定価￥128,000▶特価￥ 95,000

アイテック
- ●ITX-640(40MB/28ms)…定価￥158,000▶特価￥ 83,000
- ●ITX-680(80MB/20ms)…定価￥198,000▶特価￥ 97,000

プリンター(ケーブル・用紙付) (送料￥1,000)

- ■CZ-8PC5-BK NEW …定価￥ 96,800▶特価￥70,000
- ■CZ-8PK10…定価￥ 97,800▶特価￥71,000
- ■CZ-8PG2…定価￥160,000▶特価価格はTEL.!!
- ■CZ-8PG1…定価￥130,000▶特価価格はTEL.!!

モデムコーナー (送料￥1,000)

■COMSTARZ CLUB24/5
(NEC) 定価￥39,800
特価￥26,500 (送料・消費税込み ￥28,325)

■MD-24FB5V
(オムロン) 定価￥39,800
特価￥27,400 (送料・消費税込み ￥29,252)

P & A 特選パソコンラック (送料無料)移動自由(キャスター付)

中古パソコン(セットはモニター付) 送料￥2,000

- ●X68000セット…▶￥180,000
- ●X68000 ACEセット…▶￥200,000
- ●X68000 ACE-HDセット…▶￥215,000
- ● EXPERTセット…▶￥230,000
- ● EXPERT-HDセット…▶￥265,000
- ● PROセット…▶￥250,000
- ●X68000PRO-HDセット…▶￥270,000
- ● EXPERTⅡセット…▶￥250,000
- ● EXPERTⅡ-HDセット…▶￥320,000
- ● PROⅡセット…▶￥240,000
- ● PROⅡ-HDセット…▶￥310,000

中古パソコンはP&Aにおまかせ!!

その場で高価現金買取り・高価下取りOK!!

■まずはお電話下さい。
03-3651-1884、FAX:03-3651-0141

■下取り・買取りでお急ぎの方、直接当社に来店、または、宅急便にてお送り下さい。

●下取りの場合…価格は常に変動していますので査定額をお電話で確認して下さい。(差額は、P&A超低金利クレジットをご利用下さい。)

●買取りの場合…現品が着き次第、2日以内に買取り金額を連絡し、振込み、又は書留でお送り致します。

●近郊の方は、P&A本店まで、直接お持ち下さい。即金にて、￥1,000,000までお支払い致します。

《便利な超低金利クレジットをご利用下さい》

●月々￥1,000円からOK!! ●ボーナス払いOK(夏冬10回までOK)
●支払い回数 1回～84回 ●お支払いは、8ヶ月先からでもOK!!

アフターサービス万全
全商品保証付、専門の担当者がお客様の立場で対応します。
初期不良、輸送トラブルetc.
万が一初期不良、輸送トラブルが発生しました際には、即交換させていただきます。

●定休日/毎週水曜日＝第3水曜・木曜は連休とさせていただきます(祭日の場合は翌日になります)

通信販売お申し込みのご案内

〔現金一括でお申し込みの方〕
●商品名およびお客様の住所・氏名・電話番号をご記入の上、代金を当社まで、現金書留でお送りください。(プリンター・フロッピーの場合、本体使用機種名を明記のこと)

〔銀行振込でお申し込みの方〕
●銀行振込ご希望の方は必ずお振込みの前にお電話にてお客様のご住所・お名前・商品名等をお知らせください。
(電信扱いでお振込み下さい。)

〔振込先〕住友銀行 新小岩支店
当No.263914 ㈱ピー・アンド・エー

〔クレジットでお申し込みの方〕
●電話にてお申し込みください。クレジット申し込み用紙をお送りいたしますので、ご記入の上、当社までお送りください。
●現金特別価格でクレジットが利用できます。残金のみに金利がかかります。
●1回～84回払いまで出来ます。但し、1回のお支払い額は￥1000円以上。

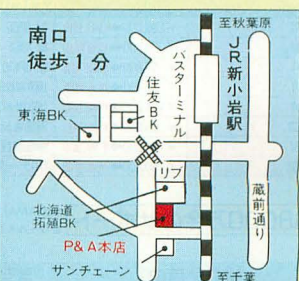
超低金利クレジット率

回 数	3	6	10	12	18	24	36	48	60	72	84
手数料	3.5	4.5	6.0	6.0	11.0	12.5	17.5	23.0	29.5	38.0	45.5

●マイコン
●ビデオ
●ビデオテープ

株式会社ピー・アンド・エー
〒124 東京都葛飾区新小岩2丁目1番地19号
☎03-3651-0148(代) FAX. 03-3651-0141

営業時間
平日:AM10:00～PM7:00
日祭:AM10:00～PM6:00

南口 徒歩1分
至秋葉原 JR新小岩駅 バスターミナル 住友BK 東海BK リブ 北海道拓殖BK P&A本店 サンチェーン 蔵前通り 至千葉

●現金書留及び銀行振込でお申し込みの方は、上記商品の料金に3%加算の上でお申し込み下さい。詳しくは、お電話でお問い合せ下さい。

［各店のお休み］
5月のお休み／2日㈭・9日㈭・16日㈭・23日㈭・30日㈭
6月のお休み／6日㈭・11日㈫・12日㈬・13日㈭・14日㈮・20日㈭・27日㈭

ワールドインアオヤマにおまかせ下さい！

INFORMATION

電話でのご注文の場合

☎ 03-3987-7771

北海道受注センター ☎011-251-6771
九州受注センター ☎092-672-7771
お好きな時間にお電話を！

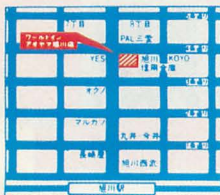
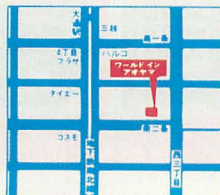
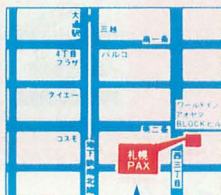
ファクシミリでご利用の場合

03-3985-5221

●ご注文方法（黒色のボールペン、またはサインペンでご記入下さい。）
①電話番号・住所・氏名又はお客様番号、お支払い方法をご記入下さい。

お客様相談室

03-3987-7795

すでにご注文いただいているお届け時間（時期）やメンテナンス、その他のお問い合せは上記へお電話下さい。

旭川店
旭川市4条8丁目ツジビル
■営業時間／11:00～19:00

札幌店
札幌市中央区南2条西3丁目リンクエギビル3F
■営業時間／11:00～19:30

札幌PAX店
札幌市中央区南2条西2丁目ブロックビル6F
■営業時間／11:00～19:30

池袋店ソフト店
豊島区東池袋1-28-6 パールシティビル2F
■営業時間／11:00～19:00

池袋本店
豊島区東池袋1-28-1
■営業時間 11:00～19:00

福岡店
福岡市中央区渡辺通り4-9-25 ユーテックプラザ3F・地下鉄天神駅下車3分
■営業時間 11:00～19:30

新コース

［福岡店のお休み］
5月のお休み／1日㈬・8日㈬・15日㈬・22日㈬・29日㈬
6月のお休み／5日㈬・11日㈫・12日㈬・13日㈭・14日㈮・19日㈬・26日㈬

X68000EXPERT II　X68000　Aコース

CZ-603C〔本体〕……………… ¥338,000
CZ-603D〔0.31カラーディスプレー〕…… ¥ 84,800
住友3M 5'2HDブランクディスケット… ¥ 18,000
御希望ゲームソフト〔人気ソフト上記よりお選び下さい。〕 サービス

定価合計 ¥440,800➡¥295,000

¥ 8,300×48回	㊟なし ㊖なし
¥15,200×24回	㊟なし ㊖なし

X68000EV I　X68000　Bコース

CZ634C TN〔本体〕……………… ¥368,000
CZ606D TN……………… ¥ 79,800
住友3M5'2HDブランクディスケット ¥ 18,000
御希望ゲームソフト……… サービス

合計¥465,800➡現金大特価
安すぎて表示できません。
クレジットでもお申し込みできます。

X68000EV I　X68000　Cコース

CZ634C TN……………… ¥368,000
CZ605D〔Newタイプ〕……… ¥ 99,800
住友3M5'2HDブランクディスケット ¥ 18,000
御希望ゲームソフト……… サービス

合計¥485,800➡現金大特価
安すぎて表示できません。
クレジットでもお申し込みできます。

X68000EV I　X68000　Dコース

CZ644C TN……………… ¥518,000
CZ606D TN……………… ¥ 79,800
住友3M5'2HDブランクディスケット ¥ 18,000
御希望ゲームソフト……… サービス

合計¥615,800➡現金大特価
安すぎて表示できません。
クレジットでもお申し込みできます。

X68000EV I　X68000　Eコース

CZ644C TN……………… ¥518,000
CZ605D〔Newタイプ〕……… ¥ 99,800
住友3M5'2HDブランクディスケット ¥ 18,000
御希望ゲームソフト……… サービス

合計¥635,800➡現金大特価
安すぎて表示できません。
クレジットでもお申し込みできます。

X68000PRO II　X68000　Fコース

CZ-653C〔本体〕……………… ¥285,000
CZ-602D〔0.39カラーディスプレーテレビ〕… ¥ 99,800
住友3M 5'2HDブランクディスケット… ¥ 18,000
御希望ゲームソフト〔人気ソフト上記よりお選び下さい。〕 サービス

定価合計 ¥402,800➡現金特価

¥ 7,200×48回	㊟なし ㊖なし
¥13,100×24回	㊟なし ㊖なし

X68000PRO II　X68000　Gコース

CZ-653C〔本体〕……………… ¥285,000
CZ-603D〔0.31カラーディスプレー〕…… ¥ 84,000
住友3M 5'2HDブランクディスケット… ¥ 18,000
御希望ゲームソフト〔人気ソフト上記よりお選び下さい。〕 サービス

定価合計 ¥387,800➡¥242,000

¥ 6,500×48回	㊟なし ㊖なし
¥11,900×24回	㊟なし ㊖なし

X68000をはじめソフト&周辺機器類は、当社池袋店・札幌店・旭川店・福岡店にて実演中です。各店X68000コーナーが常設されております。

X68000ソフト&周辺機器

品名	価格	品名	価格	品名	価格
SCSIボード(CZ-6BSI)	¥ 29,800➡現金特価	BF-68PRO	¥ 15,500➡¥ 16,800	Communication PRO-68K	¥ 19,800➡現金特価
システムサコム MIDIボード(SX-68M)	¥ 19,800➡¥ 15,300	CZ-6TU	¥ 25,000➡現金特価	Stationary PRO-68K	¥ 14,800➡現金特価
LANボード	¥268,000➡¥201,000	オムロンMD-24FP4 II	¥ 38,800➡¥29,800	DATA PRO-68K	¥ 58,000➡¥ 43,500
RS-232Cケーブル(平行)	¥ 7,200➡現金特価	オムロンMD-24FP5 II	¥ 42,800➡¥ 33,000	BUSINESS PRO-68K	¥ 68,000➡¥ 51,000
RS-232Cケーブル(クロス)	¥ 7,200➡現金特価	ローランドMT-32	¥ 64,000➡¥ 54,400	NEW Printshop PRO-68K	¥ 19,800➡現金特価
インテリジェントコントローラ	¥ 23,800➡¥ 18,900	Hyperword	¥ 39,800➡現金特価	グラフィックライブラリ vol.1	¥ 8,800➡現金特価
トラックボール	¥ 13,800➡¥ 12,000	CYBERNOTE PRO68K	¥ 19,800➡現金特価	グラフィックライブラリ vol.2	¥ 8,800➡現金特価
ジョイカード(延長コード付)	¥ 3,200➡¥ 2,900	C compiler PRO-68K	¥ 44,800➡¥ 33,600	Musicstudio PRO-68K ver1.1	¥ 28,800➡¥ 21,600
CZ-8BS1(X-1用)	¥ 23,800➡現金特価	CARD PRO-68K	¥ 29,800➡現金特価	MUSIC PRO-68K〔MIDI〕	¥ 20,500➡現金特価
拡張I/Oボックス	¥ 88,000➡現金特価	CARD PRO システム手帳リフィル集	¥ 9,800➡現金特価	ソングライブラリ 101曲集	¥ 8,800➡現金特価
アンプ内蔵スピーカーシステム	¥ 28,500➡現金特価	CARD PRO 活用フォーム集	¥ 9,800➡現金特価	Sampling PRO-68K	¥ 12,500➡現金特価
システムラック	¥ 44,800➡¥ 35,800	SX-WINDOW ver1.0	¥ 6,800➡現金特価	SOUND PRO-68K	¥ 15,800➡¥ 11,500

X68000シリーズ周辺機器

品名	価格	品名	価格	品名	価格
CZ-8NS1	¥188,000➡¥141,000	CZ-8PC5	¥ 94,800➡現金特価	I/Oデータ 2MB増設RAM	¥ 50,000➡¥ 36,500
CZ-6BN1	¥ 29,800➡現金特価	IO-735X	¥248,000➡現金特価	I/Oデータ 4MB増設RAM	¥ 88,000➡¥ 64,000
CZ-6VT1	¥ 69,800➡¥ 52,400	CZ-8PK10	¥ 97,800➡現金特価	GP-IBボード	¥ 59,800➡現金特価
CZ-6BV1	¥ 21,000➡現金特価	1MB増設RAM(CZ-600C専用)	¥ 35,000➡¥ 28,000	増設用RS-232Cボード	¥49,800 ➡現金特価
CZ-6PV1	¥198,000➡¥148,500	1MB増設RAM	¥ 28,000➡¥ 22,400	ユニバーサルI/Oボード	¥ 39,800➡現金特価
CZ-8PC3	¥ 65,800➡¥ 39,000	2MB増設RAM	¥ 60,000➡現金特価	数値演算プロセッサ	¥ 61,000➡現金特価
CZ-8PG1	¥130,000➡¥ 97,500	4MB増設RAM	¥138,000➡¥107,000	FAXボード	¥ 79,800➡¥ 55,800
CZ-8PG2	¥160,000➡現金特価	I/Oデータ 1MB増設RAM	¥ 25,000➡¥ 18,000	MIDIボード	¥ 26,800➡¥ 20,300

X68000万全のサポート
AOYAMAにて購入のX68000は万一故障の場合でも全国どこでも出張サービスがうかがいます。万一の場合ワールドインアオヤマサポート係にお電話下さい。お客様のお名前と電話番号だけで手続きは完了。

組合せ自由	激安金利にキャンパスクレジット	ゆっくり、お支払いは8ヵ月先から
各コース以外の組合せもコースをベースに周辺を合せたセット……お支払いだって御希望のパターンをお組みいたします さあ、ご相談もお見積りも受注センターもしくは各店へお気軽に	手続きカンタン、大学生の為の超低金利クレジット 20歳以上の学生の方は原則として保証人様には連絡いたしません。	クレジット業界最低の金利を有効に使って、支払いは最長8ヵ月後から始まるクレジットでも。

これは？と思ったら……どんどんお電話下さい！

グレー限定お買得セット　X68000　新Uコース

CZ-603CGY〔本体〕……………… ¥338,000
CZ-606D〔カラーディスプレー〕……… ¥ 79,800
住友3M2HDブランクディスケット……… ¥ 18,000
御希望ゲームソフト〔上記ソフトよりお選び下さい〕 ¥サービス

定価合計 ¥450,800➡¥295,000

¥ 6,100×72回	㊟なし ㊖なし
¥ 6,900×60回	㊟なし ㊖なし
¥10,300×36回	㊟なし ㊖なし
¥14,900×24回	㊟なし ㊖なし

一歩ふみこんだアートの世界　X68000　アール Rコース

CZ.604C〔本体〕……………… ¥348,000
CZ-602D〔0.39カラーディスプレー付ディスプレー〕… ¥ 99,800
住友3M2HD ブランクディスケット…… ¥ 18,000
御希望ゲームソフト〔人気ソフト上記よりお選び下さい〕 ¥サービス

定価合計 ¥465,000➡¥316,000

¥ 2,800×60回	㊟¥30,000 ㊖なし
¥ 5,000×36回	㊟¥40,000 ㊖なし
¥ 7,800×60回	㊟なし ㊖なし
¥11,900×24回	㊟¥30,000 ㊖なし

信頼と安さのツクモ ―― 卸販売も致します。☎03(3253)5599 担当/荒井

テクノパラダイス TSUKUMO

掲載商品2万円以上送料無料(離島を除く)

夏まで待てない！貴方のためにボーナス一括払受付中！(金利・手数料なし!!)

X68000シリーズ新製品登場

X68000 XVI 快速16MHz

■X68000 XVI(CZ-634C-TN)
標準タイプ…………定価¥368,000
■X68000 XVI-HD(CZ-644C-TN)
HD内蔵タイプ……定価¥518,000

●CPUクロック周波数スピードアップ(16MHz)実質約1.8倍。●増設メモリ本体内蔵可能(8MBまで)。
●NEW SX-WINDOW搭載。

X68000シリーズ 特価販売中！

■X68000PROII(CZ-653C)………定価¥285,000
■X68000PROII-HD(CZ-663C)…定価¥395,000

TSドライブいよいよ5月末日発売！

X68000シリーズ専用3.5インチフロッピーディスクドライブ
TS-3XR1
定価¥44,800

●1ドライブタイプ。
●3.5インチ2DD/2HD対応ドライブ使用。
●ユーティリティソフト付属。(ディバイスドライバ)

ツクモ特価¥35,800
(消費税別途¥1,074)

安心 迅速 高額

買い取りのツクモニューセンター店

ツクモ買い取りセンター
好・評・買・い・取・り・中・！

電話受付(AM11:00〜PM5:00)
(03)3251-9977
FAX受付(24時間)
(03)3251-0299

一流メーカー 増設メモリーボード

1MB増設RAMボード
(ACE/PRO/PROIIシリーズ用)
ツクモ特価¥17,500
(消費税別途¥525)

2MB増設RAMボード
特価¥34,800(消費税別途¥1,044)

4MB増設RAMボード
特価¥61,500(消費税別途¥1,845)

※計測技研のメモリーボードも取扱っておりますので、価格についてはお尋ね下さい。

MIDI♪コンピューターミュージック

Aセット		Bセット	
●CM-32L	¥69,000	●CM-64	¥129,000
●SX-68M	¥19,800	●SX-68M	¥19,800
●MusicstudioMu-1 Ver1.4	¥19,800	●Musicstudio Mu-1 Ver1.4	¥19,800
合計定価	¥108,600	合計定価	¥168,600

Aセット ツクモ特価¥88,000
(消費税別途¥2,640)
クレジット例(18回払・税込)
初回¥7,223+月々¥5,600×17回

Bセット ツクモ特価¥138,000
(消費税別途¥4,140)
クレジット例(24回払・税込)
初回¥7,603+月々¥6,900×23回

※「Misicstudio PRO68K Ver2.0又は「Music PRO68K」<MIDI>のソフトの場合には、¥9.500プラスになります。また、これらのソフトウェアがバージョンアップにより価格が変更になった場合には変更となります

ローランド 追加オプション機器
ステレオマイクロモニター CS-10………定価¥17,000
MIDIキーボードコントローラー PC-200 定価¥36,000
はなうたくん CP-40…………定価¥33,000

ビジネスツール

■Hyper WORD……………定価¥39,800
■Multiword NEW……………定価¥32,000
■FIXER Ver4.0………ツクモ特価¥15,800
(消費税別途¥474)
■CARD PRO-68K Ver2.0 NEW 定価¥29,800

アートツール(ハード)

■A4サイズカラーイメージスキャナー……………台数限定特価¥128,000
(消費税別途¥3,840)
■ファインスキャナーX68 HGS-68……ツクモ特価¥31,800(消費税別途¥954)
■CZ-6VT1 カラーイメージユニット 定価¥69,800
■CZ-6BV1 ビデオボード………定価¥21,000
■CZ-8PC5 48ドットカラー漢字熱転写プリンター NEW……………定価¥96,800

アートツール(ソフト)

■CANVAS PRO-68K…………定価¥29,800
■Z's STAFF PRO-68K Ver2……ツクモ特価¥46,400(消費税別途¥1,392)
■マジックパレット……ツクモ特価¥15,800
(消費税別途¥474)

開発ツール

■C Compiler PRO-68K Ver2.0……………定価¥44,800
■XBAS TO C CHECKER PRO-68K……………定価¥ 9,800

電子手帳

■ハイパー電子システム手帳 PA-9500
定価¥48,000ツクモ特価¥43,000(消費税別途¥1,290)
■スタイリッシュ電子システム手帳 PA-X1
定価¥29,800ツクモ特価¥26,000(消費税別途¥ 780)
■Telepotion PRO-68K NEW 予約受付中

ツクモグローバルカード 入/会/者/受/付/中/！

18才以上なら学生でもOK！

国内・外で大活躍
使って便利、持ってて安心！ツクモグローバルカードは、ジャックス・VISAとの提携カードです。ツクモ各店でのお買物がらくらくできるうえに、国内はもとより海外での分割ショッピングもOK！
お申し込みは☎03-3251-9898又は各店店頭で！

★各店頭では、JCB・日本信販・DC・セントラル・マスター、他各種カードも取り扱っております。

X68000用ハードディスク

—大容量記憶装置—

TX-80
●80MB SCSI/SASI両対応
定価¥108,000 ツクモ特価¥88,000
(消費税別途¥2,640)

TX-130
●130MB SCSI対応
定価¥138,000 ツクモ特価¥110,000
(消費税別途¥3,300)

TX-180
●180MB SCSI対応
定価¥185,000 ツクモ特価¥148,000
(消費税別途¥4,440)

(カラーはブラックかグレーになります)

※SCSIハードディスクとしてお使いの場合、本体がSUPER/XVI以外の場合にはSCSIボード(CS-6BSI)が必要です。

ツクモ通販センター フリーダイヤル受注専門 **0120-377-999**
商品についてのお問い合せは各店店頭又は…☎03(3251)9911へ

★表示価格には消費税は含まれておりません。

ツクモは「スーパーX PRO SHOP」です。

PRO STAFF ツクモ

パソコン本店 荒井 / N・C店 福地

九十九電機株
〒101-91 東京都千代田区神田郵便局私書箱135号

ツクモパソコン本店2F ☎03-3253-5599 (担当/荒井)

便利で安心な通信販売
ツクモ通販センター☎03-3251-9911

■ツクモニューセンター店 ☎03-3251-0987(担当/福地)
■ツクモAV/カメラ館B1 ☎03-3254-3999(担当/川名)
■ツクモ5号店 ☎03-3251-0531(担当/ 森)
■名古屋1号店 ☎052-263-1655(担当/吉高)
■名古屋2号店 ☎052-251-3399(担当/横山)
■ツクモ札幌店 ☎011-241-2299(担当/田口)

★商品のご注文は在庫確認の上お願いします。

カード払い	全国代金引き換え配達	クレジット払い	現金書留払い	銀行振込払い	各種リース払い
通信販売での御利用カード、ツクモグローバルカード、VIPカード、セントラル、ジャックス※御本人様より電話で通信販売部へお申し込み下さい。	お申し込みは☎03-3251-9911へお電話1本！配達日の指定もできます。	月々¥3,000以上の均等払いも頭金なし、夏・冬ボーナス2回払いも受付中！	〒101-91東京都千代田区神田郵便局私書箱135号 ツクモ通販センター Oh!X係	事前に☎でお届け先をご連絡下さい。富士銀行 神田支店(普)No.894047 ツクモデンキ	くわしくは各店にお問い合せ下さい。ケースに合わせてご相談にのらせて頂きます。

全国通販

SHARP 認定 PPO-SHOP **O.A.ランド**

(TEL) 03-3770-8855

■アフターサービス万全のサポート体制
●下取・買取は電話で見積りしております。責任を持って下取りさせて頂きます。

営業時間
平日………AM10:00～PM8:00
土日・祭日…AM10:00～PM6:00

▶5・15～6・14

SHARPのことなら なんでもおまかせ!!

大徳買セール!安く値切ってネ。(本体セット:送料・消費税込み)
お電話下さい。㊙価格をお知らせいたします。

流通事情により、広告表示価格は、お安くなる場合がありますので、ドンドンお電話下さい。

CYBER STICK
■CZ-8NJ2 (定価¥23,800) OAランド特価 ▶¥18,000

電子手帳 ●見やすい漢字4桁表示!! 情報任時代の必需品!!
■PA-9500(¥48,000)…▶特価¥38,000
■PA-8500(¥28,000)…▶特価¥15,000
■PA-7500(¥22,000)…▶特価¥12,000

SHARP X68000シリーズセット(送料・消費税込み)

X68000XVI NEW

①CZ-634C-TN+CZ-613D-TN
定価合計¥503,000

12回	TEL下さい	24回	TEL下さい
36回	TEL下さい	48回	TEL下さい

②CZ-634C-TN+CZ-606D-TN
定価合計¥447,800

12回	TEL下さい	24回	TEL下さい
36回	TEL下さい	48回	TEL下さい

■CZ-634C
特価¥TEL下さい!!
■CZ-644C
特価¥TEL下さい!!

X68000XVI-HD

①CZ-644C-TN+CZ-613D-TN
定価合計¥653,000

12回	TEL下さい	24回	TEL下さい
36回	TEL下さい	48回	TEL下さい

②CZ-644C-TN+CZ-606D-TN
定価合計¥597,800

12回	TEL下さい	24回	TEL下さい
36回	TEL下さい	48回	TEL下さい

■CZ-6BE2B(2MB増設メモリ/チップ型)・CZ-6BP2(数値演算プロセッサ/チップ型)も、大特価で販売中!! …………………… TEL下さい!!

X68000SUPER

①CZ-604C-TN+CZ-613D-TN
定価合計¥483,000

12回	TEL下さい	24回	TEL下さい
36回	TEL下さい	48回	TEL下さい

②CZ-604C-TN+CZ-606D-TN
定価合計¥427,800

12回	TEL下さい	24回	TEL下さい
36回	TEL下さい	48回	TEL下さい

■CZ-604C
特価¥TEL下さい!!
■CZ-623C
特価¥TEL下さい!!

X68000SUPER-HD

①CZ-623C-TN+CZ-613D-TN
定価合計¥633,000

12回	TEL下さい	24回	TEL下さい
36回	TEL下さい	48回	TEL下さい

②CZ-623C-TN+CZ-606D-TN
定価合計¥577,800

12回	TEL下さい	24回	TEL下さい
36回	TEL下さい	48回	TEL下さい

X68000PROII

①CZ-653C+CZ-613D
定価合計¥420,000

12回	TEL下さい	24回	TEL下さい
36回	TEL下さい	48回	TEL下さい

②CZ-653C+CZ-605D
定価合計¥400,000

12回	TEL下さい	24回	TEL下さい
36回	TEL下さい	48回	TEL下さい

③CZ-653C+CZ-606D
定価合計¥364,800

12回	TEL下さい	24回	TEL下さい
36回	TEL下さい	48回	TEL下さい

■CZ-653C
特価¥TEL下さい!!
■CZ-663C
特価¥TEL下さい!!

X68000PROII-HD

①CZ-663C+CZ-613D
定価合計¥530,000

12回	TEL下さい	24回	TEL下さい
36回	TEL下さい	48回	TEL下さい

②CZ-663C+CZ-605D
定価合計¥510,000

12回	TEL下さい	24回	TEL下さい
36回	TEL下さい	48回	TEL下さい

③CZ-663C+CZ-606D
定価合計¥474,800

12回	TEL下さい	24回	TEL下さい
36回	TEL下さい	48回	TEL下さい

上記組合せのディスプレイ(モニター)変更自由!!
詳しくは、お電話にてお問い合せ下さい!!

■期間中、セットでお買い上げの方には、①ジョイカード(連射式)と②テトリスやドルアーガの塔などの入ったゲームパックをプレゼント!!

周辺機器コーナー 電話で値切ろう。

プリンターセットコーナー

①CZ-8PC5 NEW 定価¥96,800
●48ドット ●熱転写カラー 漢字プリンター
大特価¥70,000

②CZ-8PK10(24ピン漢字プリンター136桁)
定価¥97,800…特価¥71,000

③CZ-8PG1(24ピンカラー漢字プリンター80桁)
定価¥130,000…特価¥93,000

④CZ-8PG2(24ピンカラー漢字プリンター136桁)
定価¥160,000…特価¥114,000

OAランド特選品.!!

■IO-735X(定価¥248,000)
●カラーイメージジェットプリンター
特価¥177,000

X68000用ハードディスク

■SCSIタイプ
●アイテック
①ITX-80S (¥128,000)……特価¥ 88,500
②ITX-130S(¥158,000)……特価¥108,000
③IT-SS40 (¥138,000)……特価¥ 70,000
●テクノジャパン
①PD-50GS (¥116,000)…特価¥ 81,000
②PD-100GS (¥148,000)…特価¥101,000
③PD-130GSX(¥168,000)…特価¥114,000
■SASIタイプ
●ロジテック
①SHD-40(¥99,800)……特価¥ 60,000
※X68000SUPER/XVI以外の機種では、SCSIボードが必要となります。
★SCSIボード………特価¥ 22,000
★光ディスク…………特価¥320,000

X68000用周辺機器コーナー

①CZ-6VT1(カラーイメージユニット)
定価¥69,800……特価¥ 52,500
②CZ-8NS1(カラーイメージスキャナー)
定価¥188,000…特価¥138,000
③CZ-6BM1(MIDIボード)
定価¥26,800……特価¥ 20,500
④CZ-6BV1(VIDEOボード)
定価¥21,000……特価¥ 15,600
⑤CZ-6TU(RGBシステムチューナー)
定価¥33,100……特価¥ 25,000
⑥CZ-64H(増設ハードディスク)
定価¥120,000……特価¥ 89,000
⑦CZ-6EB1(拡張I/Oボックス=4スロット)
定価¥88,000……特価¥ 66,000
⑧CZ-6BP1(数値演算プロセッサボード)
定価¥79,800……特価¥ 60,000

《計測技研》増設メモリ&プロセッサ

●高速増設メモリと数値演算プロセッサが一つのボードになった.!!●

● KGB-X68 PRK-00(¥34,000)…特価¥26,000
● PRK-01(¥58,000)…特価¥43,500
● PRK-02(¥74,000)…特価¥55,500
● PRK-03(¥98,000)…特価¥73,500
● PRK-04(¥122,000)…特価¥91,500
● PRK-10(¥72,000)…特価¥54,000
● KGB-X68 PRK-11(¥96,000)…特価¥ 72,000
● PRK-12(¥112,000)…特価¥ 84,000
● PRK-13(¥136,000)…特価¥102,000
■KGB-X68PRK-14(¥160,000)…特価¥115,000
■MC6888 1RC16(¥38,000)……特価¥ 28,500

I・Oデータ増設RAMボード

■PIO-6BE1-A (1MB) 定価¥25,000 特価¥17,500
■PIO-6BE2-2M (2MB) 定価¥50,000 特価¥34,500
■PIO-6BE4-4M (4MB) 定価¥88,000 特価¥59,900

■OAランド推奨ソフト

Ⓐ Telepotion PRO 68K (CZ-258BS) NEW
特価TEL下さい!!
Ⓑ Musicstudio PRO 68K Ver.2.0 (CZ-261MS) NEW
定価¥28,800
特価TEL下さい!!
Ⓒ Card PRO 68K Ver.2.0 (CZ-253MS) NEW
特価TEL下さい!!
Ⓓ Multiword PRO 68K (CZ-225BS) NEW
特価TEL下さい!!
Ⓔ CZ-245LS (C-コンパイラII)
定価¥44,800
特価¥33,500
●その他 TEL下さい!!

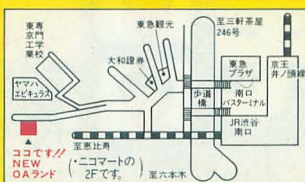
低金利クレジットをご利用下さい。平日AM10時～PM8時、土日・祭日AM10時～PM6時迄ガンバッテま～す!!

■ゲームソフト、ビジネスソフト2割、3割引きは、当り前で～す。お問い合せ下さい。

通信販売のご案内

全国通販

■銀行振込で申し込みの方は商品名及びお客様の住所・氏名・電話番号をお知らせ下さい。
〔振込先〕第一勧業銀行 渋谷支店
普通No.1163457 ㈱オーエーランド

■現金書留で送金されるお客様は電話番号と商品名、数量を明記して同封して下さい。■クレジットでご購入を希望される方は申し込み用紙をお送り致しますのでご記入の上返送して下さい。20才以上の方は、原則として保証人不要です。クレジットは1～60回払で月々5,000円よりご自由に設定できます。

■年中無休です.!!

クレジット表

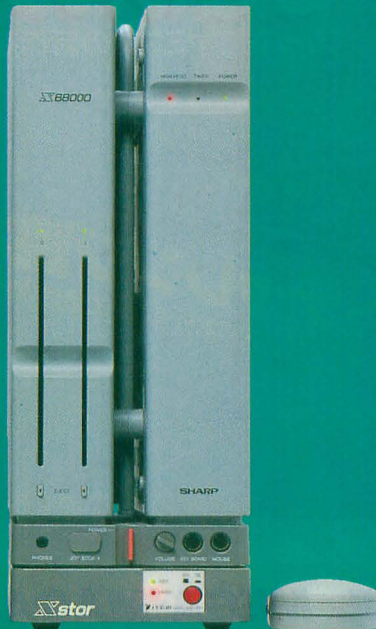
3回	3.5%	6回	4.5%	10回	6%	12回	6%	15回	8.5%	18回	11%	20回	12%
24回	12.5%	30回	17%	36回	17.5%	42回	22.5%	48回	23%	54回	29%	60回	29.5%

〒150 東京都渋谷区桜丘町3-13 アルカディア2F
☎(03)3770-8855
関東エリアの送料は、1個につき¥1,000です。 FAX(03)3770-7080

★全商品保証書付。専門のアドバイザーが、お客様のニーズに対応します。
★初期不良・輸送トラブル等に迅速に対応し、即交換させていただきます。

■表示価格は、税別表示です。詳しくは、お電話にて、お問い合せ下さい。掲載の価格は、4月下旬現在です。

Xstor

SHARP X68000専用ハードディスク

HXD040(1台目用外付モデル)

Xstor40はシャープX68000専用に開発されたハードディスクです。目的に応じて外付タイプ(2モデル)と内蔵タイプ(1モデル)を用意。従来の汎用サブシステムにはない数々の特徴とハイセンスなデザインを実現した省スペースタイプの高品質なハードディスクです。

■平均アクセスタイム23ms。又バッファサイズ、32Kバイトを装備。満足のいく高速性能を提供。
■パーソナルには余裕の40Mバイトの記憶容量。更に増設用HXD042を付加することにより最大80Mバイトまでのディスクシステムが利用可能。
■Human 68K(Ver1.00以上)、OS9対応。既存の多くのソフトウェアがそのまま利用可能。
■交替セクタをユーザー領域から独立。しかもFormatプログラムにより自動実行。
■切電時のオートパーキングロックを採用。不意な衝撃に対しても磁気面を保護。
■高品質、低価格を実現。

HXD040:Xstor40/1台目用外付モデル……………¥118,000
(X68000/ACE/EXPERT/PRO用)
HXD042:Xstor40/2台目増設用外付モデル………¥128,000
(X68000 ACE<HD> EXPERT<HD> PRO<HD> HXD040又はHXD140の増設用)
HXD140:Xstor40/内蔵用モデル……………………¥98,000

●データ転送速度/1.5MB/S ●インターフェース/SCSI(シングルユーザ) ●交替処理/FORMATコマンドによるセクタ単位の自動交替処理 ●外形寸法/35H×155W×313Dmm(HXD040/HXD042)/135H×155W×41Dmm(HXD140) ●重量/約2.5kg(HXD040/HXD042)/約800g(HXD140)

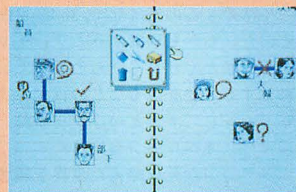
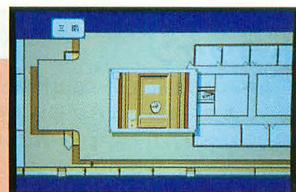
詳しいカタログが必要な方は本社までご請求下さい。
※内蔵用モデルの対応機種については、お問合せ下さい。

HXD140(内型用モデル)

株式会社アイテム
本社/〒251 神奈川県藤沢市南藤沢8-1-202
TEL.0466-27-1668(代) FAX.0466-27-2800
東京ショールーム/〒105 東京都港区新橋4-31-7中村ビル7F
TEL.03-3434-4171 FAX.03-5472-5315

SOFTWARE INFORMATION

このところ，いわゆる人気作品の発売がめじろ押しですが，今月もそれに輪をかけるかのように，移植などの情報が入ってきています。でも，なんとなく移植ばかりというのも寂しい気がするのですが……。

黄金の羅針盤
「琥珀色の遺言」でお馴染み，藤堂龍之介探偵日記シリーズ第2弾。今回の舞台となるのは，桑港航路の豪華客船・翔洋丸。その船内で白骨死体が発見され，偶然乗り合わせていた私立探偵，藤堂龍之介が捜査に乗り出す。だが，それをあざ笑うかのように新たな殺人が！ パースをつけて描かれた船内のグラフィック，混み入った演出を可能にする複数での会話，そして練り上げられたシナリオなど，推理アドベンチャーの王道を行く作品だ。（浦）

話題のソフトウェア

今月のトップは，リバーヒルの**黄金の羅針盤**。X68000版の画面写真が届きました。やっぱりキレイですねぇ。発売は6月あたりになりそうとのこと。すぐですね。

電波新聞社からは，あの**イース**がやっと登場します。画面写真は間に合わなかったけど，来月には詳しく紹介できそうです。

またまた海外からの移植ものが登場，エピック・ソニーの**ドラッケン**です。詳しくは次のページを見てね。

同じく海外移植もののプリンス・オブ・ペルシャを発売したばかりのブロダーバンド・ジャパンでは，**ループス**というパズルゲームを開発中とのこと。

さておき，発売中はシステムソフトの**キャンペーン版大戦略Ⅱ**，アートディンクの**A列車で行こうⅢ**，マキシマの**マーキュリー**，そして発売が遅れていた新声社の**スコルピウス**です。これでゴールデンウィークは安泰だね。来月詳しくレビューしますんで，お楽しみにしてちょ。

システムソフトの**大戦略Ⅲ'90**，シャープの**ダッシュ野郎**，M.N.M Softwareの**スターモビール**，ハミングバードの**ロードス島戦記〜灰色の魔女〜**は，次からのページを見てね。では，ばいなら！（古い……）

パロ強し！ お笑いの神話が始まる，か？

		(前回順位)
1.	パロディウスだ!	1
2.	遥かなるオーガスタ	初
3.	メルヘンメイズ	2↓
4.	サイレントメビウス	初
5.	マーブル・マッドネス	初
6.	エメラルドドラゴン	3↓
7.	キャンペーン版大戦略Ⅱ	−
8.	ロードス島戦記〜灰色の魔女〜	初
9.	カオスの逆襲	6↓
10.	A列車で行こうⅢ	8↓

まずは「ごめんなさい」のコーナーから。先月のチャートの三国志Ⅱと，ラグーンの初登場マークは間違いで，文章のとおりリバイバルが正解です。すみません。それから，4月号のキャンペーン版大戦略は「キャンペーン版大戦略Ⅱ」の間違いでした。よって，今月の7位は再登場ということになります。どうも申し訳ありませんでした。

さて，チャートはといえば，発売になったパロディウスだ! が，初登場の乱立で軒並みランクを落とすものが多いなか，余裕でトップをキープ。得票数は昨年のダンジョン・マスター並みです。ふぇ〜，すごいな。

じゃ初登場のハガキの声，いってみよう。

遥かなるオーガスタ：多人数で記録を競いあうのが楽しい。操作も簡単。リアルで作りがいい。T&Eの熱意が伝わってくる。スピードも文句のないレベルでよかった。

サイレントメビウス：原作のファンだから。PC-9801版で，ものすごく感動した。絵がきれいだけど，DISK枚数が気になる。ガイナックスだから。ガイナックスだから。……ズームの再来か？

マーブルマッドネス：ひさびさに熱くなれるゲームだ。トラックボールさばきが上手くなる。やっているうちに頭の中が真っ白になる。

ロードス島戦記：待たせたあげく，いきなりラプラスの魔を出すという反則をやってのけたから。出渕裕のグラフィックがいい。前からほしかったから。やっと出してくれるから。……なかなかみんなストレスが溜まってるな。

スペースがないのでまた来月だ。（浦）

ドラッケン

海外のパワフルなゲームの移植が相次いでいる。このヨーロッパはおフランス生まれのRPG「ドラッケン」もそんなゲームのひとつだ。

このゲーム，移動の仕方からして斬新なアイディアが導入されている。決まったマス目にそって移動するのではなく，360度自由に移動できるのだ。しかも移動するとプレイヤーの前に広がる光景がぐりんぐりんと動いてしまう。戦う，移動する，物を取るなどの行動もすべてアニメーションしちゃうというから新鮮。もちろんフルマウスオペレーションだ。

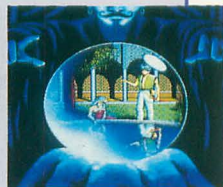
ドラゴンが死に絶えた世界を舞台に，そのドラゴンを復活させるために8つのオーブを集めるのがゲームの目的。なんかドラゴンボールみたいだな。エピック・ソニーはパソコンゲーム初挑戦だけど，クオリティの高い移植を期待したいね。（浦）

X68000用　5″2HD版　価格未定
エピック・ソニー　☎03(3475)2632

＊画面はPC-9801版のものです

プリンス・オブ・ペルシャ

APPLEⅡで発売され，PC-9801，IBM PC，AMIGAなどにも移植された人気ゲーム「プリンス・オブ・ペルシャ」。ディズニープロダクションにいた人が開発に関わったということで，そのリアルな動きがウリになっている。だが，移植された機種によって少し毛色が違っているのが面白い。たとえば，PC-9801版なら絵の美しさ，AMIGA版なら効果音のよさというところがポイントになっている。X68000に移植される際にはすべての機種のいいところがよりよくなって，完璧なものになるのではないかと期待したが，残念ながらそうではないようだ。また，3枚組によるディスク交換，2Mバイトないと効果音が出ない，そのわりにオンメモリではない，イベント前後に動きが止まる，などの問題点もある。（R.A.）

X68000用　5″2HD版　8,800円(税別)
ブロダーバンド・ジャパン　☎03(3341)1135

大戦略Ⅲ'90

超有名シリーズ，大戦略の最新版「大戦略Ⅲ'90」が発売されることが決まったぞ。先月キャンペーン版大戦略Ⅱを紹介したと思ったら，さっそく次だ。素早い攻撃だな。

この大戦略Ⅲ'90は，PC-9801版にすでにリリースされた大戦略Ⅲの改良パワーアップバージョン。お互いに兵器を生産し，敵の首都を占領するという基本ルールはそのままに，マルチスタックが可能になったり，生産の細かい指定，占領の方法の改良などが施されているのだ。行動命令には侵攻作戦，占領作戦のように大まかな指示を与えておけるから，多数のユニットの制御も大丈夫。

内容はヘビーだけど操作は簡単というこの大戦略Ⅲ'90，シミュレーションファンはチェックせずにはいられない。（浦）

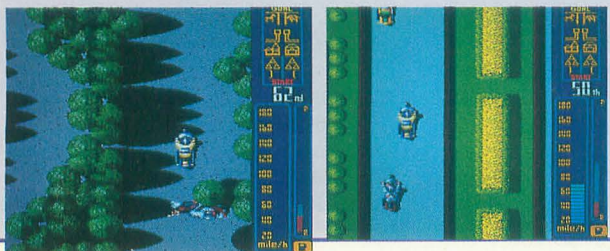
X68000用　5″2HD版2枚組　9,800円(税別)
システムソフト　☎092(752)5278

ダッシュ野郎

スカしたBGMにのってアメリカ横断のバイクレースに出よう，というわけで取り出しましたるゲームが1本。東亜プランの「ダッシュ野郎」だ。

画面はこのとおり縦スクロール型。バイクが最高速に達すると，いきなり障害物が出現してくるから，一瞬たりとも気が抜けない。ライバルのバイクは狭い道を占領するし，幅よせしてくるイジワルな奴らばかりだ。

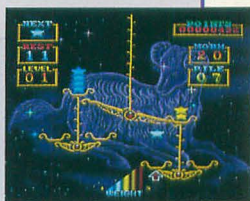
ターボエンジンを手に入れ，ガソリンを補給し，ジャンプを繰り返してひたすらゴールへ向かう。ゲームは結構ハードだけど，画面もBGMもなにやらのん気な雰囲気。最近の大作志向のゲームとはちょっと違った1本だ。（浦）

X68000用　5″2HD版　価格未定
シャープ　☎03(3260)1161

スターモビール

「スライス」「マジカルショット」とつぎつぎにゲームをリリースしているM.N.M Software。またまた新作が登場だ。その名も「スターモビール」。SFレース物じゃないぞ。星座を題材にしたパズルゲームなのだ。

まずは画面を見てちょうだい。上から降ってくる星を，この天秤の6つの皿で受けとめるのだ。もちろん，天秤だからバランスを崩すと星はこぼれて落ちちゃうし，星にも3種類の重さがあるからなかなか頭を使う。同じ色の星ではさむと真ん中の星が消えたり，8つ同じ星を積むとスペシャルボーナスがもらえたりするのが，いっそう頭を使うゲームにしているのだ。面ごとに決められた数の星を天秤に乗せればOKだが，一定数以上落っことしてしまうとゲームオーバー。

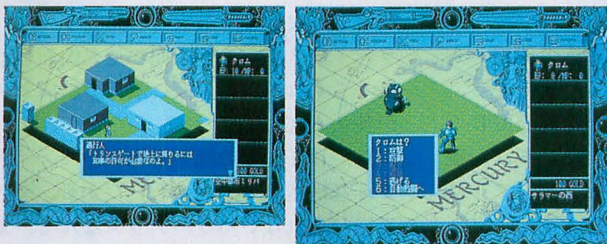
画面は見てのとおりファンシーだし，背景の星座は面が進むにつれて変わっていく。女の子でも呼んで一緒にやりたいゲームだな，これは。でへへ。（浦）

X68000用　5″2HD版　価格未定
M.N.M Software　☎0423(60)3084

マーキュリー

先月号でも紹介したマーキュリーですが，今月はもう少し詳しくゲームの進行の具合などを紹介しましょう。

まず，ぱっと見でクォータービューが目にとても鮮やかです。だからといって，AXISみたいにロボットが飛び回るゲームじゃないですよ（笑)。正真正銘のRPGです。移植の際にグラフィックなどをX68000用に新たに描き起こしたそうで，オリジナルのPC-9801版と比べると，なかなか見応えのある画面となっています。システム的にも馴染みやすいオーソドックスなものを使用しており，自動戦闘で楽ができるところは，つぼを押さえているといえるでしょう。

ちょっと難をいうと，操作性がやや悪いかもしれませんね。しかし，RPG初心者の方々には気軽に楽しめるといった点でおすすめといえるでしょう。（哲）

X68000用　5″2HD版2枚組　8,800円(税別)
マキシマ　☎06(561)2215

ロードス島戦記～灰色の魔女～

小説やOAVになってもうお馴染みの「ロードス島戦記」が，そろそろX68000にも登場するぞ。

いくつもの戦乱をくぐり抜けてきたロードス島。南に位置する暗黒の島マーモの怪物をまとめ上げた暗黒皇帝ベルドの登場によって，またまた暗雲が立ち込め始めた。

そんなある日，神官の娘レイリアが行方不明になるという事件が起こる。ドワーフのギムが捜索に出発し，戦士のパーン，ソーサラーのスレインといった仲間を加えてロードス島を旅するうちに，やがて彼らはロードス島の戦乱の中に巻き込まれていく。

スタイルは正統派のRPG。もとがテーブルトーク用のシナリオだったっていうから，話運びや世界観に期待が持てるね。（浦）

X68000用　5″2HD版　9,800円(税別)
ハミングバードソフト　☎06(315)8255

(善）のゲームミュージックでバビンチョ

とうとうドラゴンセイバーのROMを買ってしまいました。それにしても，新作のローリングサンダーⅡのほうが安いのには笑えましたね。ところで，システムⅡってミュージックテストをうまく行う方法ってないんでしょうかね。誰か知ってたら教えてね。

（お勧め度は10点満点です，念のため)

●XakⅡ＆フレイ　CD:PSCX-1016
ポリスター　2,400円(税込)

「XakⅡ」はPC-9801/8801版から,「フレイ」はMSX2からの収録。状況描写的な曲が多いので，ゲームをプレイしていないと少々ムズカシメな内容かな。しかしながら，PSGやFM音源の音色の使い方が秀逸であるので，コンピュータミュージックをやっている人には勉強になる1枚だ（パンフルートが絶品だネ)。最近のゲーム音楽に音色のオリジナル性が失われていくなかで，このマイクロキャビンサウンドを確立させたことはお見事。

・「XakⅡ」のX68000はまだですか，マイクロキャビンさぁん。

お勧め度　7

●ミスティーブルー／古代祐三　CD:ALCA-123
アルファレコード　2,000円(税込)

エニックスよりPC-9801/8801シリーズに発売されている，同名のゲームソフトのBGM集。最近，スーパーファミコンの「アクトレイザー」のBGMで業界を驚かせた，ご存じ古代祐三の作品。スキームのときのような元気一杯の曲とは違って，アニメやテレビドラマのような静かな曲調が多く，ゲームをしていないとこっちもむずかしめな内容だな。演奏音源は，例によってPC-8801のサウンドボード。比較的新鮮に感じられたのは，ドラム以外にクワイアなどの肉声音をサンプリングし，これをFMに薄く重ねたりしているところ。これはほかのソフトハウスも真似するといいかもね。

あと，おまけとして「アクトレイザー」のサウンドボードアレンジが入っているんだけど，これが悪魔城なんとかみたいでけっこう面白かった。

・収録時間50分で2,000円は安いな。

お勧め度　8

●美少女ソフトオリジナルカタログスペシャル
LD:PSLX-1002　ポリスター　6,000円(税込)

エッチソフトの総カタログが，LDになってついに発売。当たり外れの激しいこの手のソフト，買う前にこれで内容をチェックするといいかもね。なにしろ，けっこう出来がナニなソフトもちゃんと紹介してあるし，重要なシーンもみれるし，カタログとしてはかなり深く切り込んだ内容ですよ。ポリスターさん，次はLD野球拳ギャルの総カタログですかぁ？

（浦：でも1時間も見続けてると頭がばびばび)
（善：天使たちの午後Ⅰまで収録されているのは驚きだよね)
（荻：やっぱ生身がいちばん)
（純：徹夜のおともにはいいかもね)

●MUSIC FROM ボンバーマン　CD:FHCF-1104
ファンハウス　1,800円(税込)

（神奈川県の多摩梨純平さんからのリクエスト)

「MUSIC FROM ボンバーマン」です。えーと，「こんにちは，西川善司さん。いつも善バビをリンボーダンスをしながら読んでいます。CD屋を覗いたら売っていたので，思わず買ってしまいました。内容は，アレンジ3曲とPCエンジン版，ファミコン版のオリジナルサウンド，という結構月並みなものなんですが，アレンジがすごいんです。なんと，あのボンバーマンのBGMがダンスミュージック，ラップ調にアレンジされているんです。ぜひ聴いてみてください」

とのこと。私も聴いてみましたが，なるほど，こりゃすごいわ。ディストーションバイオリンはインパクトあるよね。X68000のオリジナルサウンドも入っているとよかったのにね。

・値段は1,800円と安いよ

お勧め度　7

終わりに

あれ？　今月はたった4枚だけなの？　そうなんです。ごめんなさい。今月はゴールデンウィーク進行（年末進行よりもきついといわれる）だったもんだから，結局締め切りに間に合ったのがこの4枚だけだったんですよ。そうそう，このコーナーでは，ゲームソフトのミュージックモードへの入り方など，その他ゲームミュージックに関する情報を募集してますのではがきにでも書いて送ってください。そいじゃ，また。（善）

吾輩はパロディウスである

ゲームセンターで人気を誇った「パロディウスだ!」がX68000だけに登場だ。お馴染み4種類の自キャラの中からひとつを選び，いざ発進！　全10ステージ，最高2周までプレイが可能。もちろんMIDI対応だ。

Nishikawa Zenji
西川 善司

吾輩はパロディウスである。名前はパロディウスだ。トンネルを抜ければそこは前人未踏のバビンチョ・タコワールド。あぁ，美しい……。ナンセンスがこんなにも魅力的なものだったなんて……。ああ，凄い，もうダメ，耳から鼻クソが出てきそう。

自機はドイツだ,オランダだ◆◆◆◆◆◆

イエイ，ミーはビックバイパーってんだ。江戸っ子でぇい。「し」と「ひ」の区別がつかねえぜ。いまは軍隊を退役して，たいやき屋をやっているんだがや。ちょっと，最近食いすぎちまって太っちまったでごわす。ばってん，装備はグラディウスⅠのときと同じでやんす。趣味は，7面の泡に入った雑魚女にメガホンで「明朗会計3,000円ポッキリ」とわめくことだべっちゃ。

私はタコです。私を選んだプレイヤーは，ミスして死ぬと必ず「このタコっ」っていいます。くそー。タコのどこが悪いんだ，ミスしたのはてめえじゃねえか。俺はな，グラディウスⅡのビックバイパーの装備とそっくりなんだぞ。2ウェイミサイルにリップルレーザーが強力なんだぞ。バリアはいまいち使えないけどな。趣味は，巨大化して7面のボスとキスをすることです。

ボクはツインビーです。ロケットパンチが強いと評判です。上下攻撃用に3ウェイショットもあるし，バリアを張ると障害物をもすり抜けられますよ。分身は，例によって動かないと威力が発揮できないのが，ボクの欠点といえば欠点です。趣味は，ロケットパンチで2面の「ちちびんた」のバストを触りまくることです。

やー，僕が女の子に圧倒的人気のペン太郎だよ。でもオプションは3つまでだし，スプレッドガンはベルパワーを調整しにくいし，実際パロディウスだ! をいちばん難しく遊べるキャラクターなんだ。バブルは背景もすり抜けちゃうのはツインビー君と同じ。趣味は，スプレッドガンで「チチビンタ」の股を撃って「18禁」ごっこをすることです。

どうも，私が西川善司です。誕生日は5月28日，星座はアクマイ座ー3です。昔バレンタインデーに銀紙に包まれた石コロをもらったことがあります。趣味は，グレープフルーツを縦に切って途方に暮れることです。と，それはさておき，こうして見てくると，やはりいちばん強いのはツインビーかね。その次がタコ。でも，タコはオプションを取ると，どれが自機だかわからなくなることがあるんだよね。その次にビッグバイパーで，最後にペン太郎，だなぁ。

どーするどーするドイにする◆◆◆◆◆

パロディウスだ！には，パワーゲージを全然気にしないでいいオートパワーアップと，グラディウス伝統の普通に自分の意志で行えるマニュアルパワーアップの2つがあるんだな。どっちにしようか迷うところだけれど，初心者から熟練者まで，ゆりかごから墓場まで，私はマニュアルをお勧めする。オートでは，テイルガンやダブル系統が取れないのが最大の「難」なのだ。グラディウスⅠのモアイ面や逆火山面を思い出してくれや。あの面に限って結構ダブルが使えたでしょ？　で，パロディウスだ! では，ほとんどの面が逆火山やモアイ面のような構造をしているわけだから，ダブル系統が取れないパワーアップなんて，鬼に「うまか棒」，弁慶にミヤマクワガタを持たせるようなもんなのだ。

ま，確かにマニュアルパワーアップにも欠点はある。それは，ルーレットカプセルを取ったときに，最悪，全装備を失ってしまうという危険があるからだ。オートならルーレットカプセルは出てこない。が，ルーレットカプセルの位置は固定なので，覚えてしまえばいい。もしフルパワーアップの状態で取ってしまったら「無視」に限る。いいか，絶対オートモードは使うなよ。見つけたら後ろから耳に息かけるぞ。

それと，ベルパワーは狙って取るのは3面くらいまでにすること。あ，○色ベルだ，とかいって撃つのをやめてヒヨコ編隊に激突するプレイヤーを，私はゴマンと見ている。ベルパワーでお勧めなのが，やっぱり赤と緑。赤は敵や敵弾をシャットアウトしてくれるし，緑は背景をも通り抜けできる。特に緑は後半面や2周目にはなにかと助か

X68000用　5"2HD版2枚組　9,800円(税別)
コナミ　☎03(3264)5678

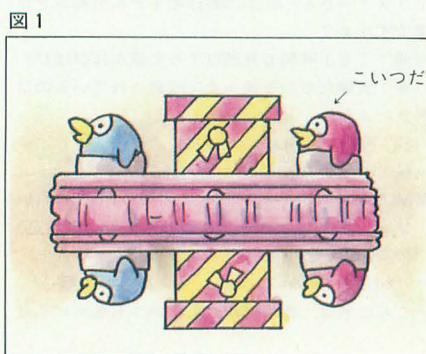
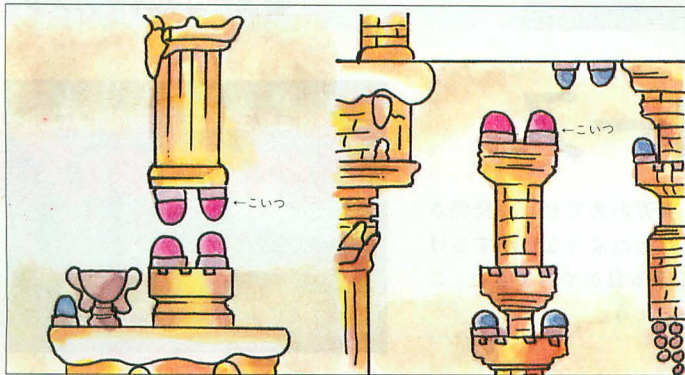
図1

こらこら，このタコはなに照れてんだよ

図2

図3

る。一定時間だけでも長生きできるのはいいよね。でも，後半はベルパワーは取れたらラッキー！　程度の心構えで進むこと。

さっそく,攻略法だい ◆◆◆◆◆◆◆◆

それではさっそく，

●ステージ1：アイランドオブパイレーツ
ガンバレ，ミスするな

●ステージ2：ピエロの涙も三度まで

ルーレットは，最初の浮遊地帯の上側のカプセルだ(図1)。ピエロはやっつけたあとは当たっても死なないぞ。チチビンタは，撃って左のバストに命中する位置にいれば，1回目の接近時にやられない。ここまでスピードアップをしないで来ると，あとが楽(イージーランクのケース，ハードではピエロ地帯で)。と，いうのはパロディウスだ！は，スピードとバリアを取れば取るほど難しくなっていく（通は「ランクが上がっていく」というらしい）のだ。イージーモードでも，ノーミスでいけば8面あたりで撃ち返し弾を吐き出してくる。だから，この辺まではスピードを取らないで来るといい。チチビンタはノースピードでもかわせるぞ，ちょっと難しいけど。

●ステージ3：お菓子城の謎

どうってことない面だが，ルーレットがトラップ的に仕掛けてあるので要注意（図2)。後半の飴玉地帯でカボチャオバケが後ろから来るが，赤ベルまたは青ベルを取ってれば楽チン。タコかツインビーなら，テイルガンでもOK。ボスは舌なめずりしたら動き出すのを覚えておこう。

●ステージ4：嗚呼，日本旅情

ここは初心者トラップの面。ダブル系統がないとちょっと難しい。まず注意したいのが，ザコのおすもうさん。ランクが上がっていると，間を入れずにフンドシビームを撃ってくる。場所は決まっているので先撃ちしよう。あとは動く桜の木。これは一度見れば余りのインパクトの強さに脳味噌に焼き付いてしまうであろうから，特筆はしない。この面も赤ベルがあると楽チンだぞ。で，火山だ。ランクが上がっていると，後ろから平気で茄子を吹き上げてくる。赤ベルパワーを噴火口に打ち込むと，茄子が落ちてこないで全部ポイントになるので美味しい。赤ベルがないときはオプションを斜めにセットしてダブルを連射。噴火の境目に一気に画面右に出る，これを繰り返す。ブタシオは，フンドシを脱ぐまでにやっつけられなかったら，画面右に上を回って高速に逃げろ。倒したあとは頭上注意！

●ステージ5：宇宙戦艦モアイ

特に難しくないが2周目は地獄。撃ち返し弾の餌食になること請け合い。さて，ここもやっぱりダブル系統が欲しい。モアイ艦長のレーザーは下にいればやられない。彼の弱点は口。目じゃないぞ，私は間違えたぞ。で，ボスモアイ（ヨシコ）がかなりに難しい。パワーアップしているなら，このボスの手前でスピードを2速にするのが無難だろう。攻略法は，図3のように緑モアイ（ヨシオ）を誘導する。なるべく同一方向に2つのヨシオを動かさないようにすること。それには右一杯に自機を持っていき，ヨシオが自分に向かってきたとこで全速で反対側（左）へ行き，ヨシコのいない(右の上または下の)場所へ行き次なるヨシオの誘導に備える。

げ，もう誌面がない？　ま，リレー連載ということで来月は別のライターさんが攻略してくれるそうだからよしとしようか。

移植の出来，異色の出来 ◆◆◆◆◆◆

パロディウスだ！がX68000に移植されると聞いたとき，私は「おい，大丈夫か，クォースのときみたいにはいかないぞ，パロディウスは！」と，華奢な胸を不安げに震わせたものだった。確かに店頭デモは重かったし，サンプル版を見て早まった某誌は「『パロディウスだ！』はXVIのために作られた？」という見出しを出してしまっていたし，おそらく「某まらさんだ」のような出来になってしまったのでは？　と思ってしまっている読者も多いことだろう。でもご安心くださーい！　パロディウスは〜，

とってもいいざきー。西条秀樹ーっ

な出来です！　動きはベリグー，音楽ベリグー，拡大縮小の特殊処理もクワイトグーってなもんで，もう，明日死ぬならこの1本級のソフトだよ〜ん。ばび！

秘技18禁ショット！　お酒はハタチになってから

音楽はMIDI対応ざんす。ぶらぼー

BGMはMIDI対応なんですが，電波新聞社の「モトス」以来の，内蔵FM＋MIDI音源が見事に融合したハイパーゲームミュージック！　ドラムスの音量が少々弱めなものの，音のぶ厚さはオリジナルサウンド以上。もちろん，ミュージックモードもあるし，もういたれりつくせりってなもんです。

ご存じの方がほとんどと思いますが，「パロディウスだ！」のBGMは，クラシックのロックアレンジがメイン。学校やなんかのイベント（運動会とか）にも合いそう。ここは一発，パロディウスでX68000を友人や先生に売り込むのもイイゾ！

総評

項目	0　5　10
ゲーム性	★★★★★★★★★★
音楽	★★★★★★★★★
グラフィック	★★★★★★★★★
スピード	★★★★★★★★
熱中度	★★★★★★★★★★
お色気	★★★★★★★

ああ，わがホームコース

Urakawa Hiroyuki
浦川 博之

X68000版「遙かなるオーガスタ」発売おめでとう。発売されると聞いてから本当にずいぶんたったような気がするけど，ともかくX68000でもプレイできる日がやってきた。これで長年の胸のつかえも取れたでしょう。

現実のゴルフを丁寧にシミュレートした，この「遙かなるオーガスタ」。知らない人でも知ってる超有名作だが，それでもよくわからない人のためにちょっと解説を。

このゲームは正式名称を「NEW 3D GOLF SIMULATION」といって，「遙かなるオーガスタ」とはそのシステム上で遊べるコースを指す。いままでのゴルフゲームといえば，真上から見た構図の2次元のもの，あるいは視点は3次元だがコースは2次元（起伏がない）のものがほとんどだった。これらはゲームとしては面白くても本物のゴルフが好きな人にとってはもの足りない。ゲーム性がまるで違っていたのだ。

じゃあ，3Dのリアルなやつを作ればいいじゃんとなるわけだけど，そう簡単に作れたら苦労はない。本物のゴルフコースをパソコン上に再現するには，コースの起伏がこうなっているという途方もない量のデータを与えてやらなきゃならない。しかも，それを高速に処理して画面に表示できなければゲームとして成り立たない。もちろん球の飛び方や転がり方の計算も複雑になる。とてつもない労力と時間をかけてねばり強く作っていく覚悟がなければとてもそんなゲームは作れるものではないのだ。

しかし，それをやってしまったのが，この「NEW 3D GOLF SIMULATION」というわけ。なにせこのゲームのためにPOLYSYSという3D図形演算パッケージを開発し，ゴルフゲームとして仕上げるのに1年半，X68000上でまとめるのにはさらに1年を費やしている。もう大きいパッケージと値段はダテじゃないというくらい気合いが入っているのだ。これだけ頑張ってくれちゃうとつい欠点も見逃してあげようかなという気にもなるけど，それはそれ，これはこれということで，気を引き締めてX68000版をチェックしていこう。

PC-9801版との違い◆◆◆◆◆◆◆◆◆

X68000版は，先に発売されていたPC-9801版から若干変更された点がある。

・当然，グラフィックは描きなおし
・プレイ時間によって背景の空の色が変わるようになった
・スタンスを変えると球がどちらに曲がるか表示が出る
・PCMを使った効果音。鳥がさえずったり，歓声やどよめく声が気分を盛り上げてくれる
・PC-9801版ではショット練習ができるだけだったが，X68000版では好きなホールに出て攻略法を研究できる

とこれだけ凝った結果，ディスク3枚組，要2Mバイトとなった。この内容ならしかたがないだろうが，ディスクの入れ替えがある分，PC-9801版よりも面倒臭くなったのも事実。ハードディスクにインストールできるバージョンアップサービスをやってほしいものだ。

憧れのオーガスタ◆◆◆◆◆◆◆◆◆◆◆

ゲームを始めるにはまずプレイヤーの登録を行う。プレイヤーごとに成績が記録され，ラウンド数やハンディキャップ，イーグル/バーディの回数や平均スコア，自己スコアのベスト5などを見ることができる。

プレイ方法は3通り。ストロークプレイ，マッチプレイ，それからトーナメントだ。メインはやはりトーナメント。ハンディキャップの判定やプレイヤー成績の登録もこのトーナメントを対象に行われる。キャディを選んだら，そこはマスターズ。

コースに出ると，視点が上空をぐるんと回ってホールの全体像を見せてくれる。しょっぱなからPOLYSYSの威力全開。スピードはPC-9801の最速モデルより速い……というわけには当然いかないが，そんな無意味な比較をしなければ快適にプレイできるだけの速さを備えている。

ショットは風向き，地面の傾斜，あとは経験と勘と現実世界の常識を考慮して決めよう。方向よし，クラブよし，スタンスよし。いくぞ。ここから先は待ったなしだ。ショットのパワーインパクトはぐるんぐるんと回るパワーゲージをクリックして決める。フルスイングをしようとよくばると，えてして失敗してへなちょこショットになってしまったりするので，8,9割のパワーでラクに打つのがコツだ。次にインパクトの位置を決めねばならない。ここでトップスピン，バックスピン，フック，スライスを打ち分ける。基本はやっぱり真ん中だ。

「スコーン」と快音を残して空に吸い込まれる打球。ボールを追いかけてどんどん画面が切り替わっていく。球の飛び方，弾み方がごく自然なのに驚くだろう。トップスピンをかければ球足が伸びるし，バックスピンをかければくくっと止まる。私が気に入っているのはトップスピンをかけたピッチ・アンド・ラン打法だ。向かい風が強いときやグリーン周辺をバンカーが取り巻

振りぬいたフォームが美しい

X68000用　5"2HD3枚組　12,800円(税別)
ティーアンドイーソフト　☎052(773)7770

いているときに，しゅるるるるるとピンに向かって転がっていく姿が勇ましい。ただ，傾斜のきつい場所などでは距離感が難しい。そもそも，この「遙かなるオーガスタ」にはコンピュータゲームにありがちな必勝法は存在しない。あるのはオーガスタというコースの攻略法だけだ。

パッティングのときはグリッド（起伏の表示）がずばばばばーっと描かれ，キャディがアドバイスしてくれる。ほかのところでもアドバイスは聞けるけど，どうも脳天気であんまり参考にならない。「ちょっと長いですけど，慎重にラインを読んで沈めちゃいましょう！」とか「ラッキー，これはイージー・バーディ，いただきですね！」とか。ミスショットでかろうじてグリーンに残ったところに「ナイス・オン！」といわれるのはイヤミ以外のなにものでもないように感じてしまう。畜生，いつか16番ホールの池に沈めてやる。

はじめのうちは，15フィート以内のパットだけを確実に沈めるように心がけたい。距離が伸びれば曲がりも大きくなるからやっぱりこれも結局は経験と勘が必要である。強ければカップをかすっていってしまったり，カップをクルッと回ってしまったり，弱ければ当然届かないので入らない。カップをどれくらい回るかまで運動方程式を解いて計算しているというから，何というかこのゲームの奥の深さには計りしれないものがあるな。

はじめのうちは80以上叩いて最下位だったりするかもしれないがメゲたりしないこと。結果はプリントアウトできるから，毎回マメにとっておいて苦手なホールなどの分析に使おう。研究に研究を重ね，何度もトライして自分の成績の変化を確かめるのが，この「遙かなるオーガスタ」の正しい楽しみ方なのだ。

マスターズ ◆◆◆◆◆◆◆◆◆◆◆◆◆◆

ちょうどこのレビューを書いている最中に本物のマスターズトーナメントが開催された。いままでゴルフに見向きもしなかった私だが，知ってるコースとなると話は別。ゲームがどれくらいリアルか確かめねばなるまいと私も見てみることにした。

なるか！ ホールインワン

チップインならバーディだ

夕暮れせまるオーガスタ

見ていてものすごく妙な気分だった。ゲームのほうを先にやって本物を見るのは初めてだと，ゲームを現実化したような気がするのだ。直線基調の「遙かなるオーガスタ」に比べて本物はきれいなカーブを描いたコースの起伏，バンカー，グリーン。テレビに向かって，「うわあ，リアルだ」とつぶやいてから，その言葉のバカバカしさに自分で苦笑してしまった。

本物のオーガスタは本当に美しかった。特に今年は天気もよく花も残っているということでいっそう華やかだった。と同時に，ゲームの画面もよくできているなと，改めて感心してしまった。マスターズ独特の華やかさ，盛り上がりといったものはないが，オーガスタの景色は十分再現されているといっていいだろう。

ただ，コンピュータのほうはグリーンとフェアウエイの色が似ているので区別がしづらい。ラフの止まり方も大きいし，ボールのバウンドも大きいような気がした。このへんはコースの攻略法にも絡んでくるので，もう少し研究してもらいたかったところだ。

それに本物の盛り上がりぶりを見てしまうと，ゲームでももっとトーナメントの華やかさを追求してくれたらなと思わずにはいられない。本物には個性的でしかも上手いゴルファーが集まっているから見ているだけでも楽しい。コンピュータのプレイヤーにも上手さや性格が設定されればまたいちだんと面白くなるんだがなあ。

ところで，オーガスタには女性キャディはいないという理由で女性キャディのグラフィックがボツになったという話を聞いたが，ニック・ファルドのキャディはブロンドのおねえさんだったぞ。はて？

感想 ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆

「遙かなるオーガスタ」の第1の美点に余計なものをつけていないことがあげられる。周辺機能といえばプレイヤーの成績を保存するデータベース程度で，あくまで正確なゴルフのシミュレートに力を注いでいる。安っぽい演出にウンザリさせられることもなく，ひたすらスコアメイクに集中することができるのだ。

いいシミュレーションゲームでは実際に行ったかのような経験とカンを養うことができる。そして，本物を見たときにそこで何が起きているかを容易に理解できるようになるものだ。

そういう意味でマスターズを観ていて彼らトッププロが何を考えてショットしているか，彼らがいかにすごい技術の持ち主であるか実感できてしまうこの「遙かなるオーガスタ」は，非常に優れたシミュレーションだということができるだろう。本物のゴルフから抜け落ちている部分はあるが，それでゴルフというスポーツのバランスが崩れてはいない。

このゲームで何回かコースを回ってからマスターズのテレビ中継を観ると，ひょっとすると解説者よりも本物のオーガスタに詳しくなっているんじゃないかと思うほど。いままでのゴルフゲームでは考えられなかったことだよね。

シューティングバリバリのゲーム少年にも，駅のホームで傘を振ってるオジサンにも勧められる必携の1本だ。

ちょっと追伸

なんとマニュアルはX68000専用にわざわざ書いてある。感激。中身もとてもよくできている。操作方法からゴルフのイロハ，こういうテクニックは考慮しているとかいないとかという話まで，上級者から初心者まで誰でもわかりやすいように書かれている。それから，オーガスタ・ナショナルの専属ゴルファーによるアドバイスのついたコースガイドもついてくる。写真も豊富でマスターズトーナメントのパンフレットとしても十分使えるデキだ。

総合評価	0　　5　　10
本物に似てる度	★★★★★★★★★★
奥の深さ	★★★★★★★★★★
グラフィック	★★★★★★★★★
サウンド	★★★★★
熱中度	★★★★★★★★★★

船のロシア人は故郷を捨てる?

Ogikubo Kei
荻窪　圭

セピアで統一された美しいグラフィックのインテリジェンスアクションゲーム。おもに，人と対話することで進んでいくゲームなので，相手の性格まで把握することが必要なのです。値は張るけどCD付きです。

ノスタルジアを手にした私は，我が家の初代X68000のスリットにシステムディスクを挿入し，くわえたばこで赤いスイッチを押す。ブート。タイトルが出る。ややもすると，ディスケットが吐き出され，ゲームディスク2を挿入しろ，という。入れる。動かない。動かないぞ。おーい。

編集室に出かけ，マシン室のX68000で片っ端から試してみる。初代でだけ動かなかった。これはまずいぞ。

というわけである。"4/23以前のノスタルジアは初代では動かない"。つまり，4/23以降のノスタルジアなら，初代でも安心だ。もし，初代X68000ユーザーで「ノスタルジアが動かないよう」という人は，メーカーに電話して新しいノスタルジアに交換してもらってくださいな。以上，情報コーナーでした。

さあ，話を戻すぞ。

＊　＊　＊

PIXY -99Xの電源を入れる。DSPはMOVIEモードだ。ディスプレイをトレシーで拭う。リラックスできるゆったりした椅子に座り，膝の上にはジョイスティック，ないしはキーボード。お茶とお茶菓子を用意して（アール・グレイティーと，スコーンか？)，それからはじめよう。適当な娯楽映画でもレンタルしてきたと思って。

気張らずに，ゆっくりとね。

X68000用 5"2HD版4枚組 11,800円(税別)
タケル ☎03(3839)1013

日本・英吉利・露西亜

船長曰く「時代も変わったもんだ。目にみえねえものが幅をきかすようになりやがる」。

映画のようなものだ。アドベンチャーゲームではない。ただ，流れるストーリーのなかで，ときどき，決心のつかない登場人物を後ろから押してやったり，尋問の手伝いをしてやったりする。決心の手伝いをするのは主人公のカスケだけではない。ときには船長であり，ときには別の男だ。

画面はつぎつぎと切り替わり，プレイヤーは主人公の知らないことも知っている。こんなところも映画的だ（ただし，推理ものでこれをやると，いろいろと不都合な矛盾が生じてしまう。ノスタルジアもその例に洩れない)。普通のアドベンチャーゲームでは，そんなことはない。プレイヤーが意志決定に関与するのは主人公だけ，プレイヤーが見られるシーンも，主人公とともにある場面だけ，というのがノーマルだからだ。

ゲームするぞ，って気張って臨んではいけない。もっとゆっくりと，字幕でも（禁則処理くらいしてほしいが）読みながら，船の爆発でも眺めているのがいい。

イリュっていう女がいる。インディアンポーカーをする。「ノスタルジア」が売りにしている「いいたいことは態度で示せ」システムの練習用だ。カーソルキーやジョイスティックのレバーで，カスケ君が身を乗り出したり，引いたりする。

稼いだにしろスったにしろ，それはそれでかまわない。

そこから物語は始まる。セピア色のスクリーンで，豪華客船の爆発が起き，船は浮かぶ鉄の牢獄。ポセイドンアドベンチャーかってんだ。そのわりに客に緊迫感がないのは，みんな，能天気だからだ。

時は1907年。日露戦争が1904～1905年。日英同盟が1902年。アメリカの大統領はルーズベルト。ノスタルジア号はアメリカの船。船長はアメリカ人。ヤマダカスケは日本人(変な名前)。アッシュビーはロイド保険から来た船にかけた保険金の支払い者のひとりでイギリス人。イリュは絵を描く女の子できっとロシア人。世が世である。日本とイギリスとロシアの三角関係ってものを見逃してはならない。日本とロシアの仲が悪かったと同様，イギリスとロシアの仲も悪くて，イギリスと日本は同盟を結びながら，対等の関係ではなかった（そういえば，日本って，外国と対等の関係で条約を交わしたことって，あったっけ)。

実在の人物もいろいろと出てくる。ストラヴィンスキーはロシアの現代音楽の作曲家で，カール老はカール・グスタフ・ユングっていう誰でも知っている心理学者。ココ・シャネルってのもきっと。

爆発・ルージュの伝言

カスケ曰く「キライだ。感性ひとつで生きようなンてヤツは……」。

画面下1/3は台詞。右上の枠は，その下に表示されているプレイヤーの独り言なり，行動の選択なりが示される。メッセージ欄に地の文が出ることはない。プレイヤーが読むのは常に会話であり，それ以外はグラフィックから読みとってくれ，ということだ。絵を見ながら，「あの野郎，どーしてあれに気がつかないんだ」って歯嚙みすることもまたしばしば。だから，プレイヤーは，

爆弾で死すものはダイイングメッセージを残す

ただ画面を眺めている時間がもっとも多い。そういうゲームだから，ストーリーと同じくらい，演出が重要となる。

アニメーションを見せるテンポやタイミング，SE，BGM。「ノスタルジア」にはサウンドのon/offや，メッセージスピードの切り替えってものがまったくない。すべてが揃って初めて完成品なのだ，と信じているのだろう。

船に仕掛けられた爆弾は，「大日本帝国のいわば秘密兵器」だった。商社マンのカスケ君は，あわれ，日本人というだけで，犯人扱いされ，濡れ衣をはらそうと，事件解決に乗り出すよくあるパターン。ご苦労様。船長室の窓にはルージュの脅迫状。

偽証・国家機密・スパイ◆◆◆◆◆◆◆◆

ルーティ曰く「真実と現実は違う」。

演出といえば，あの，売りになっている，「いいたいことは態度で示せ」システムだ。身を乗り出したり，脅したり，すかしたり，困ったり，黙ったり。

そもそも，人と会話するとき，「尋ねる」/「調べる」/「聞く」なんて考えるだろうか。会話というものは駆け引きである。アドベンチャーゲームのシステム上の欠陥はその選択方式にあった。何かを見るときは視線のシミュレーションを，会話するときは態度のシミュレーションをせねばならなかったのだ。

ノスタルジアの「いいたいことは態度で示せ」システムは態度のシミュレーションを行うものである。

選択できる態度は，その時々の状況によって異なるが，最大13種類。身を乗り出したり引っこめたり，立ち上がって迫ったり，頭を抱え込んだり。しかも，「身を乗り出す」とかいうのが並んでいて，そこから選ぶのではなく，カーソルキーを動かすとリアルタイムでカスケ君の動きも変わるから，自分がどういう選択をしようとしているかは一目瞭然なのだ。いい加減，「～を聞く」とか「～について尋ねる」なんて選択肢にはあきあきしていたのだ。

このブラックってのはただのいやなヤツ

これが緊張の爆弾解体シーン

しかも，完全ストーリー主導型だから，こちらが全部尋ねるまで進行が待っていることはない。ただし，同じ台詞が2度続くようなマネはなしにしてほしい。なりそうなときは，その選択肢だけなくしてしまってもいいから。

そういうわけで，無限ループとも思われる虚しい行為もなし。実にスムーズに電気紙芝居を楽しめる。裏切り，共謀，偽称，偽証，国家機密，殺人。この手の映画にあってノスタルジアに足りないものといえば，格闘と情事だけだ。(情事くらいは)入れておいてほしかったが，まあ，いいや。無理やりいろんな要素を詰め込んでもろくなことになりゃしないから。特に，日本の娯楽映画。アメリカの真似して，娯楽映画に恋愛を挟み込むのだけは禁止したい。日本人の描くくそったれな恋愛は，娯楽映画とは無縁のものだ。あーいうのは，飯を食うように恋愛するアメリカ人に任せておけばいい。それができないのなら，娯楽は娯楽に徹するべきなのだ。

さて，プレイヤーのやることは，選択のあと押しと，「いいたいことは態度で示せ」システムだけではない。もうひとつ。「爆弾解体」だ。マスターキートンにでもなった気分で，慎重にやろう。緊張感あふれるところだ。これ以上ばらしちゃうとだめだからばらさないけど，勝手にストーリーが進むなか，急に大事なところを任されると，緊張する。

女は魔物・男は策謀◆◆◆◆◆◆◆◆◆◆◆

ルーティ曰く「ひとは，自分がみたいようにしか現実を解釈しません」。

映画的だという意味では，「闇の血族」よりずっといい。でも，ハードボイルドしようとして勇み足しているところが少なからずある。ハードボイルドってのは，もっと屈折していて，硬派なものだ。ふざけた物言いは屈折の結果であって，カッコつけではない。

映画的だという意味では，「最後までいくのが異様にむずかしい」ってこともない。最後にちょっと××があって，つまずくと，ひと足先にかっこいいエンディングを拝む(ゲームオーバーとはいわない。これもひとつの終わり)。解くものではなく，ストーリーと緊張感を楽しむゲームだから，これで，いいのだ。

映画的だという意味で演技を見れば，ブラックってやつの秘書がなかなかいい。自分の能力に頼りたいと思いながらも，結局は女としての自分に頼ってしまう，この手の女をよく表現している。例によって，主人公は大根。怪力ロシア人は演技過剰。ついでに，女は魔物，的な描かれ方をしている。どうして女は魔物かというと，論理的矛盾を矛盾と思わないからだ。いつの世も。それが故に，男は女を理解できないのだ。

3×3に並んだマスがポイント。最大で13個。左にいくほど態度がでかく，右は弱気になる。さらに上下左右が加わる。この写真は一番でかい態度。カーソルは3×3のマスの左下に

物語は勝手に進む

プレイヤーはさながら観客だ。それが悪いといっているのではない。こういうのも娯楽の一形態として認めるのにやぶさかではない。巨大迷路へ遊びにいって，そこで椅子に座らされて自動的にぐるっと一周して「はい，終わり」ってやられると「ムカッ！」っとくるけど，ディズニーランドでジャングルクルーズに乗って「思いどおりに進んでくれない」といって怒るヤツはいないのと同じ。

つまり，何日もかけて，エンディングを見ることを信条とするゲーマーには勧められないが，もっと気楽に，映画を観賞するつもりで（映画にしては高価だが）遊ぶにはちょうどいいゲームだ，ということ。

四畳半で9ボールを

「遥かなるオーガスタ」がゴルフを３Ｄでリアルにシミュレートしたゲームなら，こちらはビリヤードを３Ｄでリアルにシミュレートしたゲーム。キューをマウスに持ち替えて，おちついた雰囲気を楽しもう。

Kageyama Hiroaki
影山 裕昭

数年前，ビリヤードが全国的に流行した時期があった。あちこちにビリヤード場，プールバーが乱立し，週末ともなると盛り場のビリヤード場では，１時間待ちくらい当たり前だった。しかし，だ。しだいに世間の人々はストレス発散の場をカラオケボックスへと移していき，ビリヤード場は次々にカラオケボックスへと見事な変身をとげるのであった。

マジカルショットはビリヤードゲームに３Ｄ表示を取り入れた意欲的な作品だ。おかげで従来の２Ｄ表示では不可能だった「視点を変えて手玉を見る」といったことが，リアルタイムで行えるようになった。ビリヤード台を真上，あるいは真横から表示した昔のゲームと比べると，玉がどのような配置で台の上に置かれているか，いろんな角度から確認できるようになったわけだ。もちろん，３Ｄ表示になったことで臨場感も抜群。コンピュータビリヤードゲームに新風を吹き込んだM.N.M Softwareに拍手を送りたい。

本質的な部分

まずは画面写真を見ていただきたい。まるでレイトレーシングで描かれた画像のように見えるが，これがマジカルショットのゲーム画面。いままで発売されたどのビリヤードゲームよりも，リアルで美しい表示だということがわかるだろう。台の上の玉は単色のベタ塗りでなく，自然な陰影がつけられている。これだけのことで，玉の丸みや厚みをリアルに感じることができるんだから面白いもの。

ビリヤードにはいろいろな遊び方があるが，マジカルショットで遊べるのは９ボールゲーム。ひとりで遊べばポケットするコツを徹底的に練習することができるし，友達と対戦プレイをすることもできる。また，コンピュータを相手にして個性ある８人の選手と対戦することも可能。

画面がリアルなことは写真でわかったが，さて，内容がどこまでリアルなものかと案ずる方もいるかもしれない。しかし，玉同士の衝突などを含めたすべての玉の動きは，実際の玉の動きと比べて特におかしいと感じるところがないし，台はポケットの角まで再現されている。手玉のコントロールについては，引き玉，ひねりと完全にシミュレートされているが，惜しむらくはマッセ，ジャンプボールができないこと。ゲームだからこそ，実際にはとうていできない技も駆使したいと思うのは僕だけではないだろう。とはいえ，それ以外では玉の動きが完全にシミュレートされ，基本的な手玉コントロールも実現されているので，納得のいくレベルではある。

そのほか，バンキングやブレイクショットの練習まであって，本気でこのゲームを極めたい人にはありがたいものになると思う。特に，ブレイクショットの練習は玉の動きを見ているだけでも楽しめる。

面白いのが直前のショットを再現する，インスタントリプレイ。視点はコンピュータがランダムに決めるのだが，自分で視点が決められないというのにはちょっと疑問を感じる。ビリヤード台を見る視点をファンクションキーなどに割り当てて，自分で視点を決められるようにするのが柔軟な発想だと思うのだが……。

マウスをうまく使え

操作はフルマウスオペレーション。画面右下にあるメニューにはブレイクショットや相手のファールで手玉を好きな位置に置くときに使うMOVE，手玉を突く方向を決めるROTATE，手玉に押し玉やひねりを設定するADJUSTがある。SHOTではマウスを下に動かすとキューを引き，上に動かすとキューを押し出す。玉を突く力はマウスを動かす速さで決まる。

実際に遊んでみると，操作の大部分がROTATEとSHOTの繰り返しであることがわかると思う。欠点としては，SHOTの下にEXITが並んでいるので，EXITをSHOTと間違って選択してしまうんじゃないかと（EXITを選択すると確認メッセージが表示されるが）たいへん気を使う。

このゲームの特徴はこれまでもいってきたように，台をいろいろな角度から見た様子をリアルタイムに表示する３Ｄ処理。視

X68000用　5"2HD版　7,800円(税別)
M.N.M Software　☎0423(60)3084

チャイニーズがかわいい

ブレイクショットはいつも気分いい

注）DOCTOR2を組み込んでいる場合は，OPT.1を押しながら起動してください。

点を変えるにはROTATEを選択しているときにマウスを左右に動かすと，手玉を中心にしてビリヤード台が左右に回転し，上下に動かすと視点が高くなったり低くなったりする。つまり，視点は手玉を中心にして半径ｒの球面上を動いていき，画面に表示するわけ（わかるかな？）。

これは非常にかっこいいのだが，画面書き換えが遅い。マウスを横にちょっと動かすと，0.3秒くらいの間があって，画面が書き換えられる感じ。率直にいうと，ゲームをやるには耐えられないスピードである。

しかし，これを解決する方法が存在する。かなキーをロックすると台のポケット部分が表示されなくなり，画面にはラシャ（台に貼られた緑の布）と玉だけが表示されるモードになる。マウスを動かすと，グルングルンと面白いようにラシャが回転する。もちろん上に乗っている玉も一緒に。

どうやら，こちらのモードで突く方向を決定して，突く直前にかなキーのロックを解除するのがマジカルショットの正しい遊び方のようだ。なぜか，マニュアルでは触れられていないのだが，これを読んだあなた，お試しあれ。

9ボールのルール ◆◆◆◆◆◆◆◆◆◆

さて，コンピュータと対戦するときは3回戦勝負。前述したとおり対戦相手は8人。

手玉に右上ひねりをかけてみる

かなキーをロックしたところ

しかし，初めて遊ぶときは4人しか選択できない。勝負に勝つことによってしだいに残る4人と対戦できるようになるようだ。試合は本格的にバンキングで先攻後攻を決めるので，バンキング練習の成果が表れるところ。バンキングに勝つとブレイクショットの権利を得る。

ここで9ボールを知らない人のために簡単にルールを説明しよう。自分は白玉（手玉と呼ぶ）を突いて，1から9まで番号のつけられた玉のうち最小番号の玉（的玉と呼ぶ）から順に当ててポケットに落としていき，最終的に9ボールをポケットした人が勝ちというもの。ただし，マジカルショットでは玉の番号までは表示されないので，画面右のNEXTと書かれたところに次に当てるべき色の玉が表示されるという仕組みになっている。なお，ポケットした場合は続けてショットすることができるが，次のような場合はファールとなり選手交替となる。

1）手玉が最初に的玉に当たらなかった
2）手玉がポケットした
3）手玉が台から飛び出した
4）的玉には当たったがポケットせず，なおかつ手玉，的玉がクッションに当たらなかった

さて，手玉の位置を決めたら，玉を突き出す方向を決め，マウスを上下に動かして

もうひと工夫ほしいインスタントリプレイ

ゲーム前にはこんなシーンも

力強く手玉を突く。手玉が玉に当たると玉と玉が弾け合う音が響き，菱形にきれいに並べられた玉があっという間に四方八方に飛び散る。9色に塗り分けられた玉が台の上をかけめぐる。玉の動きはかなり速い。この時点で9ボールがポケットしたら，いきなり1勝になる(ブレイク9という)。まあ，ブレイク9は狙って出るものではないだろうけど。ゲーム中で使われる効果音はサンプリングしたものを使っているらしく，場を盛り上げてくれる。玉と玉が弾け合う音や，ポケットしたときの音がたいへんよろしい。効果音がいいとBGMを流さないでゲームをしてみたいと思うのだが，残念なことにBGMのON/OFFの切り替えはできない。残念だ。

うーん，惜しい ◆◆◆◆◆◆◆◆◆◆◆◆

対戦モードでは高校生のお兄ちゃん，お嬢さん，プロレスラー，熟女などが相手をしてくれる。対戦相手には個性があって，たとえば，イワン・フロスキーとかいうプロレスラーのおっちゃんならバンキングからパワー全開のビリヤードを展開するし，お嬢さんは上品なビリヤードをする。これはこれでいいのだが，個性を会話で出そうとしていて，はずかしいセリフを吐いたりする。ゲームに水を差すことになりかねないし，コンピュータを対戦相手にして遊ぶとセリフがじゃまで僕にはどうしても耐えられない。ひとりで遊ぶのもたまにはいいが，夕飯をおごったりして友達を強引に誘ってでも，対戦で遊びたいゲームだ。

肝心の部分はいいだけに……

何度もいうが，手玉を見る視点を変えられることがこのゲームの目玉。かなキーをロックすれば操作性もいいし，もともと玉の動きは悪くないのでシステムとしての完成度はなかなかのものだと評価する。しかし，こうなるとほかの部分で嫌なところが露骨に表れてくる。いちばん気に触るのが，対戦相手をコンピュータにしたときに表示される対戦相手の顔のグラフィック。お世辞にもうまいとはいえず，瞬きさせるくらいならほかの処理を軽くすることを考えてほしいもんだ。おまけに，この対戦相手が喋る喋る。画面にセリフが表示されるだけだが，はっきりいってじゃまなイベント。このソフトを買う人は純粋にビリヤードを楽しみたい人だろうから，こんな演出はあってもうっとうしく感じるだけだと思う。どうせやるなら，もっとシミュレートの部分を徹底的に凝ったビリヤードゲームにしてほしかった。インスタントリプレイの不徹底，BGMのON/OFF設定がないところ，シラけるセリフ。このあたり，いまひとつ詰めが足りなかったのではないだろうか。

総合評価	0　　5　　10
ゲームシステム	★★★★★★★
セリフ	★★★
効果音	★★★★★★★★★
熱中度	★★★★★
プールバー再現度	★★★★★★★

AFTER REVIEW

今回は横スクロールシューティングゲーム2本。EXACTのデビュー作「NAIOUS」と老舗ソフトハウス，ウルフ・チームの「ソル・フィース」です。どちらも技術的に素晴らしいものでしたね。

NAIOUS

▶新潟県から，新参入のソフトハウスが，横スクロール（縦もあるが）シューティングゲームというX68000得意の（よくできたゲームがすでに存在する）ジャンルのゲームを発売。このことを聞いて，期待に胸をふくらませた人はいただろうか。あまりいなかったかもしれない。上記のような決していいとは思えない条件のもと，しかも，雑誌などに掲載される写真も見栄えはよくなかった。しかし，実際に動いているのを見てビックリ，プレイしてみてまたビックリ。なんだ，よくできているじゃない。“新参入にしては”という感情が心の隅に残っていたから，とも考えられるが，プレイしたときの爽快感はタダモノでないものを感じさせる。世間ではラスタスクロールがすごいなどと技術面ばかりに評価が偏っているが，私はこの爽快感を評価したい。でも，あとのほうの面は少し難しすぎて，爽快感がなくなっているかな。

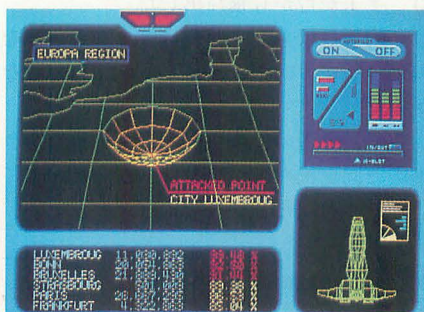
早川　富士雄(27)滋賀県

▶色はちょっとケバイような気もするけど，このゲームの特徴はやはりラスタスクロールなどに代表されるグラフィック関係のエフェクトですよね。面と面の間に挿入されるワイヤフレームのデモもかっこいい。さらにゲーム中には巧妙なトラップの数々。と，いいところばかり書いてみたけど，決していいところばかりということではありません。やはり，ツメが甘い。そのため飽きやすくなっている。新しいソフトハウスだからといえばしかたがないともいえますが，プロなんだから批判されるのは当たり前。その批判をバネに次のゲームを開発してほしいです。

柄本　章吾(19)山形県

▶このゲームの技術に関してはいまさらほめてもしょうがないのだが，やはり技術力はほめないわけにはいかないだろう。自分たちができることは全部やっているという印象を受け，非常に好感が持てるのだが，このゲームは，技術だけでは解決しえない問題も同時に教えてくれている。

たとえば，グラフィックの配色の組み合わせや背景の形状といったものが，ゲームの難易度に寄与しすぎているという状態は，難易度が安易に固定されてしまうので，好ましくないといえるし，さらにそれに加えて敵のパターン性がきびしく，いきおいプレイにも正確なパターンが要求されてしまうところも，ゲームを難しくしているだけなので避けてほしいと思う人は少なくないはずである。

技術力があることは非常によいことであるのだが，何事も「過ぎたるは及ばざるがごとし」というわけで，ツボを心得た使い方のなされた，気負いのないスマートな次回作を期待したいと思う。　(N.Y.)

ソル・フィース

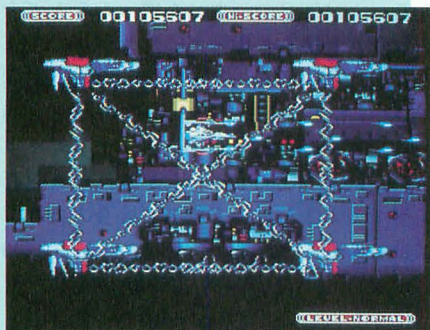
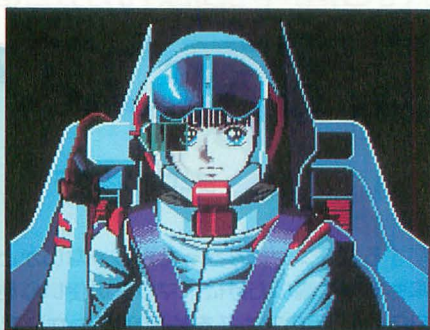
▶まず，オープニングのアニメーションに驚かされる。驚くと同時にいろいろな意味で「うーん」というため息を発してしまう。いいんだけれど，でもねえという感情を含んだため息である。やっぱりみんなあのテのアニメーションが好きなのかなあ。しかし，こういうのばっかりっていうのも……。そして，ゲーム内容。ちょっと自機が大きくて弾に当たりやすいかな，と思いつつ先へと進んでいくと，スプライトは回転するわ，関節はうねうねするわ，すごすぎてわけのわからない仕掛けがいっぱいあるわで大騒ぎ。ああ，すごい。パワーアップなども目新しく，いろいろとアイデアが盛り込まれていて，なかなか楽しいんだけど，前に出した「グラナダ」のすごさが頭にあるので「もう少しいろいろな点でがんばれたのでは？」とも思ってしまう。

井上　孝志(17)栃木県

▶はじめて画面を見たときには，あまり面白そうだとは思わなかった。でも，実際に遊んでみると，結構，バランスよくまとまっている。理不尽に難しい部分はほとんどなく，死んでもパワーアップアイテムが出てくるので，はまりも少ないのが気に入った。回転スプライトなど技術的にも感心させられるし，難易度もちょうどいい。問題はBGM，もう少しがんばってくれたらよかったのに。　木下　徹(17)北海道

▶いいなあ，この趣味にはしりまくったオープニング。さすがウルフ・チーム，こういうことには心血注いでがんばっているのがよくわかる。ゲーム自体も面白いんだけど，キャラクターデザインをもう少ししっかりしてほしいな。しかし，6面のボスはしつこすぎるぞ。西田　利也(19)神奈川県

▶ソル・フィースは見せることに徹したソフトであるといえる。どの場面にも作者が見せたいと思っているキャラクターが登場するため，視覚的に飽きのこないゲームであるといっても過言ではない。

が，ゲーム性となるとパターン重視の構成になっており，なかなかハードな攻撃が繰り広げられていくので，結果的に敵を早め早めに倒していかないと先に進めなくなっている。すると，敵の動きなどを鑑賞する余裕はどこかにいってしまい，最後にはボスの誘導レーザーを，紙一重で避けようとする自分を発見することになってしまうのが現実である。

結局のところこのゲームは，うまい人のプレイを鑑賞しつつ，回転し踊りまくるキャラクターを盆栽のように愛でるだけで満足できてしまうのである。やらずにはいられなくなるような，ドキドキした気持ちも一緒に与えてくれるような贅沢なゲームであってほしいと思うのだが，それは欲張りなのだろうか？　(N.Y.)

発売中のソフト

★サブナック　工画堂スタジオ
X68000用　5″2HD版2枚組　7,800円(税別)
★ファンタジーⅣ　スタークラフト
X68000用　5″2HD版　9,800円(税別)
★プリンス・オブ・ペルシャ　ブロダーバンドジャパン
X68000用　5″2HD版３枚組　8,800円(税別)
★ボンバーマン　システムソフト
X68000用　5″2HD版　7,800円(税別)
★キャンペーン版大戦略II　システムソフト
X68000用　5″2HD版2枚組　9,800円(税別)
★Ａ列車で行こうIII　ブラザー工業(TAKERU)
X68000用　5″2HD版　9,800円(税込)
★マーキュリー　マキシマ
X68000用　5″2HD版2枚組　8,800円(税別)
★スコルピウス　新声社
X68000用　5″2HD版　7,800円(税別)
★パロディウスだ！　コナミ
X68000用　5″2HD版2枚組　9,800円(税別)
★遥かなるオーガスタ　T&E SOFT
X68000用　5″2HD版3枚組　12,800円(税別)
★シューティング68K　ブラザー工業(TAKERU)
X68000用　5″2HD版　6,800円(税込)
★ノスタルジア　タケル
X68000用　5″2HD版4枚組　11,800円(税別)

新作情報

★SPINDIZZY　アルシスソフトウェア
X68000用　5″2HD版　価格未定
★ドラッケン　エピック・ソニー
X68000用　5″2HD版　9,700円(税別)
★eXOn　日本ソフテック
X68000用　5″2HD版　価格未定
★ロードス島戦記　ハミングバードソフト
X68000用　5″2HD版3枚組　9,800円(税別)
★生中継68　コナミ
X68000用　5″2HD版　価格未定
★サイレントメビウス　ゼネラルプロダクツ
X68000用　5″2HD版　価格未定
★黄金の羅針盤　リバーヒルソフト
X68000用　5″2HD版　価格未定
★シムアース　イマジニア
X68000用　5″2HD版　12,800円(税別)
★ファランクス　ズーム
X68000用　5″2HD版　価格未定
★パワーモンガー　イマジニア
X68000用　5″2HD版　12,800円(税別)
★ループス　ブロダーバンドジャパン
X68000用　5″2HD版　7,800円(税別)
★ライヒスリッター　エニックス
X68000用　5″2HD版　価格未定
★大戦略III'90　システムソフト
X68000用　5″2HD版2枚組　9,800円(税別)
★ダッシュ野郎　シャープ
X68000用　5″2HD版　価格未定
★スターモビール　M.N.M Software
X68000用　5″2HD版　価格未定

創刊9周年記念特別企画

PC-9801用マウスをつなぐ

Mounai Toshiyuki 毛内 俊行

これほどマウスを酷使するパソコンはなかったんじゃないかといわれるX68000。ソフトのマウス対応率も非常に高いだけに，マウスの使い心地には気をつかいたいものです。今回はPC-9801用のマウスをX68000につないでみましょう。

皆さんのマウスは元気ですか？ X68000はマウスオペレーションを前提とした設計がなされているので，マウスの使用頻度はかなりのものです。加えて，アフターバーナー，ダンジョンマスター，ポピュラス，シムシティー……。

私のマウスも先日，左ボタンが風邪をこじらせて死んでしまいました。しかし，X68000用のマウスはとても高価です。買いかえようと思って出かけた店では，2,500円という値札をつけた隣の棚のPC-9801用のマウスについ目がいってしまいます。値段もさることながら品揃えの豊富さも魅力です。そこで今回はPC-9801用マウスをX68000につなげるためのアダプタを製作し，みんなでルンルン気分に浸ってみようと思います。

作り方

早速アダプタの作り方を説明してしまいましょう。用意する材料は，表1に示したとおりです。材料の中に「マウス基板」とあるのは，あなたのマウスを分解して取り出した基板です。実はマウスのシリアルインタフェイス部分に，マウス基板の回路をそのまま使っているのです。このお陰でインタフェイス部分の設計も省略でき，その分のコストも安く仕上がりました。また，アダプタの心臓部というべきシリアルインタフェイス部分が，すでに完成しているので，実際の作業は基板とコネクタをコードでつなぐだけという単純な作業だけになり，誰でも簡単にアダプタを作れるようになっています。

ただし，自分のマウスを壊してしまうのですから，できれば壊れて使えなくなった（ただし基板は無事な）マウスの基板を使いましょう。だいたい，1～2年も使ったマウスなら，クリックボタンにガタがきていることと思います。また，どこも壊れていない健康なマウスでも問題なく使えますので，どうしてもアダプタを作りたい人は新品のマウスを買ったうえで古いマウスを壊して作ってください。

なお，本文で使っているマウスはCZ-8NM3（マウストラックボール）とCZ-8NM2A（X68000 PROに付属してきたマウス）ですが，X68000につながるマウスならすべて使えると思いますので機会のある方は試してみてください。私は，昔MZ-2500につないでいたCZ-8NM2を使いました。型は少し古いのですが，問題なく使うことができました。

マウスやトラックボールをこのように接続する

9ピンD-SUBコネクタとは，ジョイスティックのコネクタにも使われているコネクタなのでわかるでしょう。今回はメスコネクタですから，X68000のジョイスティック端子に差すことができるほうのコネクタです（間違っても差さないように）。

ビニールコードは，1本につき10cmもあれば十分です。これを8本用意してください。なお，本文の説明ではこのビニールコードを色分けして呼んでいます。使用している色は，白，青，緑，赤，黄，オレンジ，黒，茶の8色です。これは，私が最初にアダプタのテスト回路を組んだときに使用したジョイスティックコードの色からきています。パーツ屋へ行けば10本くらい束になってカラーのビニールコードを売っており，たいていこれらの色を含んでいるはずです。もっとも，電器回路そのものにコードの色

表1　材料

材料	数量
マウス基板	1個
9ピンD-SUBコネクタ（メス）	1個
ビニールコード（10cm）	8本
プラスチックケース	1個

TR-01N

完成したアダプタ

は関係ありませんから，なければ適当なコードで代用してください。

プラスチックケースはマウス基板が入るだけの大きさがあれば十分です。外見とは異なり，基板の大きさはマウストラックボールのほうがPROマウスよりも小さいので，当然マウストラックボールのほうが小さいケースで作ることができます。なるべく見栄えのいいケースを探しましょう。

さて，材料はこんなものです。マウス基板を除けば，材料にかかる費用は1,500円もあれば足りるでしょう。これにPC-9801用のマウスを3,000円で買ったとしても，合計4,500円にすぎません。もう，いくらマウスを壊しても大丈夫です。

また，用意する工具ですが，ハンダごて，ラジオペンチ，ニッパ，ハンドドリル，やすり，ドライバなど，ハード工作に使いそうな道具を一式揃えればいいでしょう。できればハンダごてだけは，IC工作用のものを用意してください。これはこて先からの漏れ電流が少ないので，回路を壊す心配がありません。

実際の製作は図のとおりです。まず，図1に従ってコードをハンダ付けします。図はコネクタの裏側（ハンダ付けをする側）から見たピン配置になっています。よくわからない人は，コネクタに小さく書いてあるピン番号を確認しながらハンダ付けしてください。

コネクタ側のハンダ付けが済んだら，基板の回転軸部分を取り外します。これがあるとマウス側からの信号が混信してカーソルがうまく動かなくなってしまいます。基板側の工作は使うマウスによって異なります。マウストラックボールを使う人は図2，PROマウスを使う人は図3を見てください。

軸の取り外しが済んだら，コードを基板側にハンダ付けします。このとき，ICの足元に直接コードをハンダ付けするので，ハンダごての熱でICを壊さないように素早くハンダ付けをしましょう。詳しくは図2，図3を見てください。

全部配線ができたら，ケースにしまう前に一度コンピュータにつないで動かしてみましょう。見事に動けばケースにしまって完成です。ケースへ基板をネジで固定する人は基板の配線部分をネジでショートさせないように気をつけてください。心配な人は，少し強度が落ちますが，強力接着剤を

マウストラックボールの改造例

やっぱりトラックボールがいい

図1　コネクタの機能と接続するコードの色

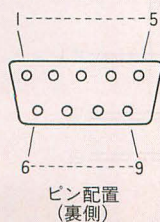
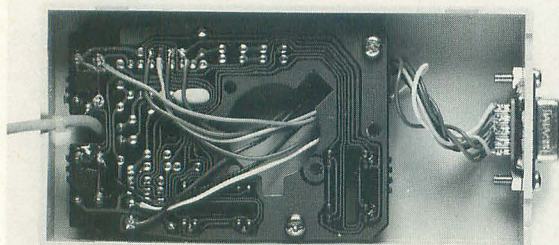
ピン配置（裏側）

番号	機能	コードの色
1	＋5V	白
2	XA	青
3	XB	緑
4	YA	赤
5	YB	黄
6	LEFT	オレンジ
7	NC	——
8	RIGHT	黒
9	GND	茶

図2　CZ-8NM3の基板の加工

1.回転軸ユニットを基板から切り離す

ユニットは，フレキシブル基板で本体基板に取りつけてあるので，接合部分にハンダごてをあてて，ゆっくりとはがしてください

2.基板の指定された箇所にコードをハンダ付けする

PRO用マウスの改造例

使いましょう。

実際に使ってみる

さて，実際に使ってみた結果ですが，これがなんと「いままでとまったく変わらないじゃあーりませんか！」なのです。そう，最初の予想以上に違和感なく使えたのです。私が使ったのはNEOSのバスマウスでしたが，その後編集部にあるEGGマウスでも快適に使うことができました。PC-9801用のバスマウスならどれでも使用可能なはずです。

さらにトラックボールでもテストしました。PC-9801用のトラックボールには，結構使いやすそうなものが出回っていて，一度試してみたいと前々から思っていました。今回使ったトラックボールは東京理化販売のTR-01Nという製品です。このトラックボールは，大きなボールと，手のひらにあわせた曲線的なデザインがとても魅力的な製品です。

しかし実際に使ってみると，マウスカーソルの移動が異常に速く，SX-WINDOW以外のアプリケーション上で使うには少し使い辛いかもしれません。試しにPC-9801にこのトラックボールをつないだときも，マウスカーソルはディスプレイ上を所狭しと飛び回っていました。これは，このトラックボールのカウント数が大きい（300カウント）ことが災いした結果なのですが，どうやら原因はそれだけでなくHumanとの相性に少々問題があるようです。X68000のOS, Human68kではマウスカーソルの移動量をマウスの移動スピードに応じて変化させているのです。

たとえば，マウスを動かすとき，さっさと動かした場合は同じ距離をのろのろと動かした場合に比べてマウスカーソルの移動量が大きくなります。簡単に確かめられるので試してみてください。

しかし，これがゲームとなれば話は別です。マーブルマッドネスではその性能をまざまざと見せつけてくれました。実際，先月のゲームレビューでは，丹氏がこのトラックボールを使ってマーブルマッドネスをやっていました。また，遊撃王IIや，アフターバーナー，サイバリオンなど，皆それぞれ，マウスを使っていたときにはできなかった微妙な操作が可能になりました（最初はちょっと慣れが必要かもしれませんが)。

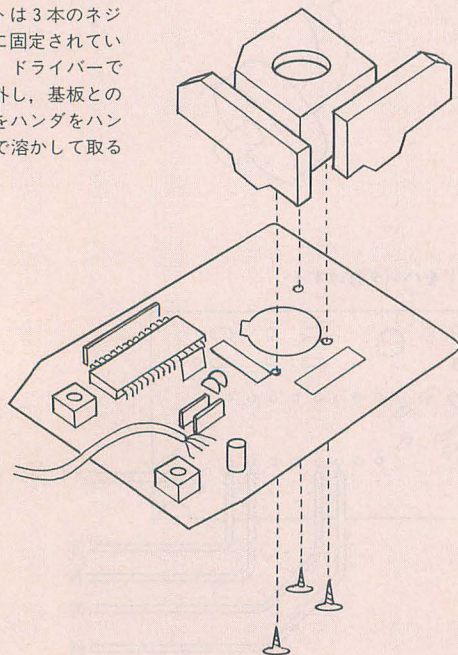
最後に

実は私は，ハードウェア工作などというものはとても苦手で，特に回路の設計から始めた装置は一度もまともに動いたものがありませんでした。今回の工作がコネクタと基板のあいだをコードで結ぶだけという簡単なものになった背景には，私のハードウェア音痴が原因だといっても過言ではありません。

しかしその結果，壊れたマウスにも第2の活躍の場が与えられ，我々もPC-9801用の安いマウスを使うことができるという一石二鳥の結果を得られたのですからよかったのではないかと思います。実際，インタフェイス部分を自分で製作した場合，製作の費用はもっともっと高くなるでしょう。X68000ユーザーならきっと壊れたマウスのひとつや2つはお持ちでしょう(?)。皆さんもひとつ，このアダプタを作ってみてはいかがでしょうか？

図3　CZ-8NM2Aの基板の加工

1.回転軸ユニットを基板から外す

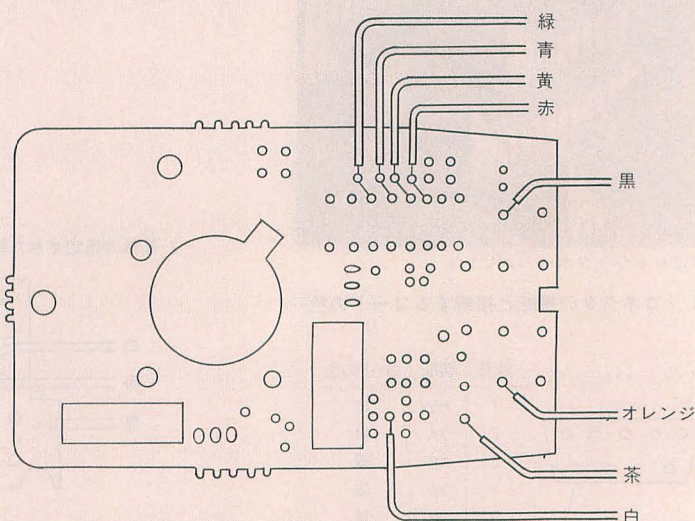
ユニットは3本のネジで基板に固定されているので，ドライバーでネジを外し，基板との接合部をハンダをハンダごてで溶かして取る

2.基板の指定された箇所にコードをハンダ付けする

特集

初心者のための環境構成術

HUMAN.SYS

X68000は自由なマシンだ。ハードウェアとソフトウェアが実に率直で自然な関係を持ち，それがユーザーに対しても奥深く開かれた環境を提供している。しかもX68000はパワーユーザーにとってだけでなく入門者にとっても，ユーザーの取り組み方に呼応した，より自由な環境を作り出すことができる。ユーザーがすすんでアクセスすれば，それに応じた結果が得られるはずだ。

今回の特集で本誌は初めて「初心者のための」という言葉をタイトルに使った。とはいえ，初心者向きとか上級者向きとかいった分け方はOh!Xが本来意図するところではない。本誌もまた読者のスタンスやアプローチによって自由な読み方のできる誌面作りを目指したいと考えている。初心者にとっての問題は上級者が次世代のために読み取らねばならない問題でもある。

CONTENTS

器なくして中身なし
まずはハードウェア環境の整備から

Ogikubo Kei
荻窪 圭

大きなポテンシャルをもつパソコンX68000。だが，
そこに秘められた力を発揮するには，もっといい環境が必要だ。
いちばん重要なのはメモリとハードディスク。
あなたのマシンはまだ能力を生かしきれていないのでは？

ジュラ紀から問われ続けながら，未だ答えのない「パソコンをいったい何に使うんだ」。だいたい，パソコンなんてのは，上記の問いに対して，言葉にできなくてもいいから，自分なりの答えを持っている人だけが使えばいいもの。問う人は使わねばいい。

しかし。パソコンなる深淵な玩具は，一度手にし，隅から隅までなでまわしてみないと，自分なりの答えが出せない。頭で考えてどうのこうのしてみても，コンピュータという新しい概念は，結局，古来の言葉を超えたものなのである。ああ，ややこしい機械よ。

って具合で，近年稀に見る真面目なオープニングをさらしてしまったわけだが，ときにはそれもまた春の嵐である。

さてと，自分がパソコンを使って何かなさんとするときの基盤。X68000の場合，いうまでもなく，ひとつの基盤がX68000本体であり，もうひとつの基盤がHuman68k，およびSX-WINDOWとなる。基盤というのがアレなら，環境といおう。

で，タイトルからわかるとおり，ハードウェア環境の話をするわけだ。

ハードウェア環境ってのは弘法も筆が滑るってなもんだ

いつのことからか，っていっても，ほとんど起源からといってもいい。パソコンの性能はCPUで語られてきた。

おいおい，時代は変わったんだよ。って，誰にいってんだか。

でも，いまだに性能はCPUで語られる。特に，価格性能比（コストパフォーマンスってやつだな）はついついCPUで語られがちだ。実態は大きく異なるのに。

おいこら，パソコン本体というのはCPUだけじゃないじゃないないか。

たとえばRAM容量であり，たとえば表示能力であり，たとえば補助記憶装置（平たくいえばFDとかHDDとか）であり，そのほかの付加機能（グラフィックとかサウンドとか）であり，さらにはキーボードであり，マウスであり，ポップアップハンドルである。

では何が重要か。コンピュータというからには，CPUとメモリと補助記憶装置である。少なくとも，X68000という世界ではサウンドとか表示能力とかは固定されているから，ほかの要素といえばこの3つだ。パソコンの使い勝手を左右する3大ポイントだ。買っちまったらしょうがないCPUはどうしようもないが，ほかの2つはあとから整えることができる。

メモリの大小は近所の公園とディズニーランドの違いのようなものだ

脳の怠慢によって，いきなり結論からいってしまうと，メモリは重要である。何よりも重要である。MS-DOSマシンユーザーはWINDOWS 3.0を使おうとして初めてそれに気づく。CPUのパワーがいくらあっても，彼が力を発揮する“場”がないとたいしたことはできない。つまり広い部屋が必要ということだ。

では，X68000にとってどのくらいが狭い部屋でどれだけあれば広い部屋か。

狭い部屋　：2Mバイト
普通の部屋：4Mバイト
広い部屋　：8Mバイト

である。SX-WINDOW登場前と後でひとつずれたという感じだな。私にとっては，1月号の付録ディスクにあった“Zs_EX”前と後で大きく変わった。

誰がためにメモリはいる

誰がためってCPUのためである。

CPUはIPLというプログラムにより，ディスクからHuman68kのシステムを読み込む。これでメインメモリのいくばくかが埋まる。しかるのちに，CONFIG.SYSというファイルをチェックする。ここにはシステムがどういう形で立ち上がればいいかが記述してあるからだ。

まず，FILESやらBUFFERSやらのためのメモリを確保する。ここでまたメモリは少し埋まる。BELLと称してビープ音代わりにサンプリングした音を組み込むと，そのサンプリングデータに応じてメモリが埋まる。

続いて，いろいろなデバイスドライバをメモリ上に読み込む。たくさん組み込むと，たくさん埋まる。

最低限必要なのは，

数値演算ドライバ FLOAT2.X

である。が，最低限の生活が保障されたとて，なにも嬉しくはない。

たいてい，次のものを必要とする。

プリンタドライバ PRNDRV.SYS

ただし，プリンタを所有していない場合はこの限りではない。

PCMドライバ PCMDRV.SYS

ただし，ビープ音しか鳴らさない場合はこの限りではない。が，こいつは組み込んでもメモリを圧迫しないので，気にせず組み込むべし。

OPMドライバ OPMDRV.X

ただし，あとから組み込むことも可。代わりにOPMDやらMUSICDRVやらを組み込むと，もっとメモリを埋めてくれる。

ここからが佳境だ。

RAMディスクドライバ RAMDISK.SYS

こいつが問題だ。指定した値だけ，ブヨ〜ンとメモリを確保するのである。こんなものを使った日にはメモリは2Mバイトでも絶対足りない。

ヒストリドライバ HISTORY.X

実は，結構メモリを食う。とりあえずなくてもなんとかなるものではある。

IOCS.X

こいつは絶対といっていいほど必要だ。気持ちの問題。

RS-232Cドライバ RSDRV.SYS

まずいらない。通信やっている人もいらない。RSDRV.SYSがなくても通信できる通信ソフトを使えばいい。複数チャンネルのRS-232Cを使う人は必需だ。

日本語FEP　ASK.SYS or FIXER4.SYS

こいつがまた100KB以上もメモリを埋めてくるんだ。これが。どっちもそうだ。

SX-WINDOW システム　FSX.X

これはSX-WINDOWしないならいらない。けど。するならいる。とってもメモリを圧迫してくれる。

まだいろいろあるけど，以下，略。

*

ユーザーはこういったデバイスドライバから必要なものを選択して，さらに欲しいものがあったら常駐させて，自分のコンピューティング環境とする。するのだ。

ここで表1を見てほしい。

私が計ったところによるので例によっていー加減である。わざわざいろんなCONFIG.SYSを作って実行してみたのだ。CONFIG.SYSには“DEVICE”の行しかないため，FILESやBUFFERSの値などでいろいろと結果は変わるが，とりあえず，こんなもんだ。

NO DRVっていうのは，CONFIG.SYSファイルがない状態で，メモリが2Mバイトある状態でのフリーエリアをMEMFREEコマンドで調べたものだ。ついでにハードディスクがつながっているのも考慮に入れよう。

なんと，1.79MBもメモリが余っている。が，逆に考えれば200KB以上もHuman68kとCOMMAND.Xがメモリ上で働いているというわけだ。

ここにいろいろとデバイスドライバを組み込んでいくわけで，下にある主なものからいくつかピックアップして，NO DRVの値から引いてやればいい。ちなみに，一番下の“FIXER4(OGI)”とあるのは，私がいつも使っているFIXERのコンフィギュレーションファイルだ。変換用のバッファを山ほど確保しているので，こんなにメモリを食っている（自分で驚いた）。だから「快適快適」とほざきながら，FIXERを使っているわけだ。

さて，計算してみよう。

よほど無茶をしない限り，なおかつSX-WINDOWを使おうと思わない限り，1.5MBくらいフリーエリアがあるはずだ。

さて，1MBしかメインメモリがないとしたらどうだ。OPMとASKを入れただけで515KBしか余らないではないか。これじゃあXC Ver2のコンパイルもできやしない。MAKEなんて夢の夢だ。

表1　デバイスドライバ占有メモリ（いーかげん）

	config.sys時	コマンドとして実行
fsx.x	288KB	286KB
float2.x(ver.2)	16KB	20KB
opmdrv.x	88KB	85KB
iocs.x	24KB	13KB
history.x	88KB	85KB
pcmdrv.sys	0KB	—
ask68k.sys	192KB	—
fixer4.sys	128KB	—
fixer4(ogikubo)	288KB	—

やっぱり4Mバイトぐらい欲しい

私は2Mバイトでも足りない。でも，まだ，GCCを使ってプログラム開発することがないからいい。某泉大介氏は，ハードディスクを買うかメモリを増設するか悩んで，結局メモリを4Mバイトにした実績の持ち主である（さすがに最近はハードディスクもつないでいるらしいが）。そのくらい，メモリは欲しいのだ。

理想をいおう。X68000上で行いたいすべての作業をひとつの環境で賄えるだけのメモリがあるのがベストだ。

たとえば，Zs_EXを使うにはフリーエリアが1.5MB以上必要だ。Cを使うなら，RAMディスクを多く確保したい。SX-WINDOWを使うなら，FSX.Xを常駐させたい。

リブートなしですべてこなそうとすると，4Mバイト必要なのだ。

私は2Mバイトしかないので，PDSに頼っている。起動時に組み込むデバイスドライバを選択できるデバイスドライバがあるのだ。

私はそれを使って，Zs_EX用，SX-WINDOW用，Hyper word用，FIXER用，ASK用などなど使い分けているのである。いちいちリセットするのである。

が，X68000の場合，メモリを増やせば解決するということだ。DOSマシンはそういうわけにはいかないからね。WINDOWS 3.0であればメモリを増やせば増やしただけ使えるようになるけど，あれでそこそこの速度を得ようと思ったら，6MB以上のメモリが必要だから。それを思えば救われる。

ハードディスクの有無は労働量を確実に減らす

メモリを2Mにしたら，次はハードディスクだ。あると何がいいか。

答え：ディスクの抜き差しの回数が確実に減る。

やはりこれではないだろうか。アクセスが速いってところも注目したいが，それ以上に，怠惰を絵に描いたようなスチャラカな私としては，ディスクの抜き差し回数が減ったのが嬉しい。システムやツールやアプリケーションは全部ハードディスクに放り込んだので，フロッピーディスクの抜き差しは，ゲームをするときとデータをセーブするときと，新しく何かをインストールするときだけになった。うーん。これは嬉しいのである。

メモリを増設したら，速やかにハードディスクを購入すべし。40Mバイトで安いのを探せば，6万円くらいで買える。

RAMを増設したり，ハードディスクを増設したときにまずすべきこと

RAMを増設する。1MBしかない人は，自分の機種にあった専用の内蔵増設RAMを買ってこよう。無印X68000クラスになるとショップを探さないと駄目かもしれないが，そういうときは取り寄せてもらうか，思い切ってXVIに買い換えてしまう。

増設はねじ回しを必要とする。カバーを開ける必要もある。後学のために，カバーを開けて中を覗いておくのもいいだろう。

さて，RAMを増やしたり，ハードディスクを接続したり，した。

次は，SWITCHコマンドの実行である。これが重要だ。

X68000はSWITCHコマンドで，内蔵しているRAMの容量や，ハードディスクの数を設定してやらないと，知らん顔を決め込むのだ。そうなったら間抜けだぞ。

では，「検討」を祈る。以下の記事には“要2Mバイト”や“要ハードディスク”といった話が少なくないからだ。

A>からのアプローチ(1)

三種の神器DIR/CD/TYPE

Izumi Daisuke
泉 大介

**A>それはX68000のOS「Human68K」の案内役である。
ここではCOMMAND.Xというシェルによって多くのサービスが受けられる。
大規模なSX-WINDOWとは違った小さくとも奥深いシステムだ。
少々とっつきにくいが，基本的3つのコマンドから使ってみよう。**

初代X68000よりずっと本体に添付されてきたビジュアルシェルは，より本格的なSX-WINDOWの登場とともに役目を終え，SCSIデバイスの採用を機にシステムディスクから姿を消すことになりました。この結果，今では標準のシステムディスクをドライブにセットして電源を入れると，本誌の付録ディスクでお馴染みのビジュアルシェルではなく，

A>

という味もそっけもない表示がユーザーを出迎えるようになっています。初めてパソコンを買い，面倒な配線もマニュアルを見ながらこなし，ワクワクしながらシステムディスクをセットして電源を入れた結果がこれです。あんまりだと思いませんか。せめて，「はじめまして私はX68000です。私はかくかくしかじかのことができます。どうぞ仕事を教えてください」くらいのことはいってくれてもよさそうなものです。

しかし，残念ながらコンピュータはまだそこまで賢くなく，そんな設定を行おうものなら，毎回毎回電源を入れるたびに「はじめまして……」とやられることになってしまいます。というわけで（かどうかは定かではないが）「はじめまして……」の分を省略し，「どうぞ仕事を教えてください」という部分だけを実行しているのが，例の「A>」なのです。

ビジュアルシェルのアイコンとマウスを基本とした操作に比べるといかにも無愛想な感じがしますが，多くの先人がビジュアルシェルに飽きたらず「A>」のコマンドモードに移ってきたのも事実です。付録ディスクに入っているビジュアルシェルは影山氏の力によって改造が加えられ，随分と使いやすいものになっています。しかし，それでもかなわないほどの柔軟な，あるいは複雑な作業が，こちらでは行えるのです。この記事は，「A>」でお馴染みのコマンドモードが拓くX68000の世界へのガイダンスです。

＊

さて，「仕事を教えてください」といわれても，どんな仕事ができるのかもわからない状態では手の下しようがありません。おもむろにマニュアルを開いて読み始めることになります。X68000付属のマニュアルはよくできているのですが，ページ数が多く，すべてを読み通して覚え込もうなどとすると確実に挫折します。そんなあなたに贈る三種の神器，これだけ覚えておけば確実に最初の一歩を踏み出せる3つの命令（コマンドと呼びます）を紹介しましょう。

ディスクの中味を見る

X68000は実にさまざまな仕事をこなせるようになっています。そのうちで主なものは，ユーザーがキーボードから指定して実行することができます。たとえば買ってきたフロッピーディスクのフォーマットを行うとか，フロッピーディスクのコピーを行うだとかいった仕事がそうです。

これらの仕事を全部覚えておくのはさすがにX68000といえど大変なので，必要に応じてディスクから取り出して実行するようになっています。ディスクのフォーマットやディスクコピーなどの仕事は，システムディスクの中に収められています。逆にいえば，ディスクの中味を見れば，どんな仕事ができるのかもわかることになります。

というわけで，三種の神器の第1はディスクの中味を見るというコマンドです。これは，

DIR

という文字で指定します。「A>」の表示に続けて，

A>dir

（大文字でも小文字でもかまわない）とキーボードでタイプし，最後にリターンキー（↵キー）を押せばOKです。

ここで，一般に使われている用語について説明しておきましょう。まず「A>」という表示ですが，これは「プロンプト」と呼ばれます。プロンプトとは促すという意味です。次に「dir」というのがコマンドの名前です。そしてキーボードで文字列をタイプしてリターンキーを押すことを「入力する」といいます。ここで上の操作を用語を使って言い直すと「プロンプトに続けてDIRコマンドを入力する」となります。なんだか難しいことをやるように聞こえますね。

実際の会話では，「DIRコマンドを入力する」なんていわずに「ディレクトリをとる」といったくだけた表現を使います。ディレクトリというのはデパートなどで各階の案内を表にしたものをいいますよね。DIRはその略なわけです。ちなみに，システムディスクをセットして電源を入れることは，

三種の神器

「システムを起動する」といいます。

さて，DIRコマンドを入力すると，画面には図１のようにディレクトリ表示がなされます（デパートの案内板と違って無愛想な文字列ですが)。図は大きく２つに分けることができます。最初の３行はフロッピーディスクの情報を表示している部分です。最初に表示されているのはフロッピーディスクの名前です。このディスクには「Human68k_Ver2」という名前がつけられています。その右に「A:¥」と表示されていますが，これについては後述します。

２行目には現在表示しているファイルの数と，フロッピーディスクに入れることのできるデータ容量1.25Mバイトのうち，現在何Kバイト使っていて残り何Kバイト使えるかが表示されています。そして３行目には現在表示しているファイルが何Kバイト使っているかが示されています。

ファイルという言葉が登場しましたね。日常生活では新聞の切り抜きや資料などを綴じておくのにファイルを使います。転じてコンピュータでは，ディスクに格納したデータを綴じておくのにファイルが使用されます。というのも，ディスクの中ではデータは空いている場所にバラバラにして格納されるため，綴じておかないとデータの続きがどこにあるのかわからなくなってしまうからです。ディスクの中のデータをどうやって綴じているのか興味を持たれるかと思いますが，あまりに技術的になりすぎますのでここでは触れません。そして，資料を綴じたファイルの背表紙に内容を表す名前を書いておくのと同じように，ディスク内のデータを綴じたファイルにも名前をつけておきます。こうしてつけられた名前（ファイル名）が，４行目以降に表示されています。

４行目以降の表示も，大きく２種類に分けることができます。「<dir>」と表示されているものと，表示されていないものです。「<dir>」と表示されているものはディレクトリです。このなかにはさらにファイルの並んだ案内図があるということです。

ファイルはディスク内にバラバラに入れられたデータを綴じるものでしたが，ディレクトリはファイルを綴じるものと考えればよいでしょう。実生活同様，ファイルを乱雑に並べたのでは目的のファイルを探し出すのは大変です。そこで，ファイルを分類し，同じようなファイルはまとめて管理しようというわけです。

ところで，ディスクのフォーマットを行ったりディスクのコピーを行うような，それらしい名前のファイルは見当たりません。これらのファイルは「X68000の仕事ファイル」として，BINディレクトリにまとめられています。DIRコマンドで表示してみましょう。

A>dir bin

のようにDIRコマンドにディレクトリ名を与えると，そのディレクトリに入っているファイルを表示することができます(図２)。フォーマットを行うファイルは「FORMAT」，ディスクのコピーを行うファイルは「DISKCOPY」です。実際にこれらの仕事を行わせるには，

A>format

A>diskcopy

と入力すればOKです。

●ファイル名の成り立ち

図１のファイル名部分にはディレクトリかどうかだけでなくさらにいろいろなデータが表示されています。どのようなデータが表示されているのかをまとめたのが図３です。

まず行頭にあるのが「ファイル名本体」で，その次が「拡張子」と呼ばれるデータです。一般に「ファイル名本体」と「拡張

図1　DIRコマンドを入力する

```
A>dir

 Human68k_Ver2          A:¥
   13 ファイル         1145K Byte 使用中        76K Byte 使用可能
  ファイル使用量         42K Byte 使用
CONFIG          SYS        469  90-05-15  12:00:00
KEY             SYS        712  87-05-15  12:00:00
USKCG           SYS       8028  87-05-15  12:00:00
BEEP            SYS       1024  87-05-15  12:00:00
STARTUP         ENV         33  90-05-15  12:00:00
COMMAND         X        28026  90-05-05  12:00:00
AUTOEXEC        BAT         40  90-05-15  12:00:00
SYS                 <dir>       91-03-15  12:00:00
HIS                 <dir>       91-03-15  12:00:00
BIN                 <dir>       91-03-15  12:00:00
BASIC2              <dir>       91-03-15  12:00:00
ASK                 <dir>       91-03-15  12:00:00
ETC                 <dir>       91-03-15  12:00:00
```

図2　BINディレクトリを表示する

```
A>dir bin

 Human68k_Ver2          A:¥BIN
   35 ファイル         1145K Byte 使用中        76K Byte 使用可能
  ファイル使用量        613K Byte 使用
ATTRIB          X          922  87-05-15  12:00:00
BACKUP          X        35250  89-02-10  12:00:00
BIND            X         7468  89-02-10  12:00:00
CHKDSK          X         2568  89-02-10  12:00:00
COPY2           X        38056  89-04-04  12:00:00
COPYALL         X         2210  90-05-05  12:00:00
CUSTOM          X        44692  90-05-15  12:00:00
DICM            X        38552  89-02-10  12:00:00
DISKCOPY        X        34578  90-06-15  12:00:00
DRIVE           X         3370  90-05-15  12:00:00
DUMP            X         1416  89-02-10  12:00:00
ED              X        35746  90-05-05  12:00:00
FC              X         3588  89-02-10  12:00:00
FIND            X         3492  89-02-10  12:00:00
FORMAT          X        95664  90-09-01  12:00:00
                :                 :
                :                 :
WHERE           X         1526  89-02-10  12:00:00
ED              HLP       5430  87-02-05  12:00:00
MENU            CNF      14472  90-05-15  12:00:00

 (途中省略)
```

図3　DIRコマンドで表示される情報の読み方

```
CONFIG          SYS       469     90-05-15    12:00:00
ファイル名本体  拡張子    サイズ   作成日付    作成時刻
      ファイル名

<dir>の場合はファイルサイズが省略される
```

子」をあわせたものをファイル名と呼んでおり，「ファイル名本体」と「拡張子」をピリオド（.）で区切って表記します。図3の例では，

CONFIG.SYS

となりますね。

図1の最初の行で「A:¥」と表示されていた部分が，図2では「A:¥bin」に変わっているのにお気づきですか。ここには，現在表示しているディレクトリ名が示されるようになっています。

最初の「A:」は「ドライブ名」と呼ばれます。ドライブ名はディスクドライブにつけられた名前で，フロッピーディスクでシステムを起動した場合，0番のフロッピーディスクが「Aドライブ」に，1番のフロッピーディスクが「Bドライブ」になります。

続く「¥」は，「〜の中の」と読みます。したがって，「A:¥bin」は「Aドライブの中のBINディレクトリ」という意味になります。BINディレクトリの中にあるFORMAT.Xを表記するなら，

A:¥BIN¥FORMAT.X

となりますね。

BINディレクトリの中を見るのに，

A>dir bin

としましたが，ディレクトリ名を正確に指定して，

A>dir a:¥bin

とやってもかまいません。

ここで少し補足しておきましょう。ディレクトリはファイルを綴じたものだと説明しましたが，ファイルだけでなくディレクトリもその中に綴じておくことができます。とすれば，図1に表示されているファイルやディレクトリもまた，ディレクトリに綴じ込まれたものと見ることができますね。この大もとのディレクトリを「ルートディレクトリ」といいます。ルートとは，根本・根源という意味です。ドライブ名に続けて「¥」を書くと，暗黙のうちにこのルートディレクトリを指定したことになります。したがって，「A:¥bin」は「Aドライブの中の（ルートディレクトリの中の）binディレクトリ」という意味になります。

カレントディレクトリを変更する

DIRコマンドだけを入力すると表示されるディレクトリのことをカレントディレクトリといいます。カレントとは「現在の」という意味です。最初は図1のように表示されますから，カレントディレクトリは「Aドライブのルートディレクトリ」になっているわけです。

CDコマンドは，このカレントディレクトリを変更するコマンドです。カレントディレクトリにあるBINディレクトリに変更するなら，

A>cd bin

のように，ディレクトリ名を指定するだけでカレントディレクトリが変更されます。もちろん，

A>cd a:¥bin

と正確に指定してもかまいません。これで，DIRコマンドだけを入力すると図2のファイルが表示されるようになります。元のディレクトリに帰るには，

A>cd ..

と入力します。「..」というのは特別なディレクトリ名で，カレントディレクトリが入っているディレクトリを意味します。この場合BINディレクトリが入っているのはルートディレクトリですから，もちろん，

A>cd a:¥

としてルートディレクトリに直接戻すこともできます。cd とは change directory（ディレクトリを変更する）を略したものです。

単に「cd」とだけ入力すると，カレントディレクトリ名が表示されます。カレントディレクトリをBINディレクトリに移して試してみてください。

再び用語ですが，カレントディレクトリが入っているディレクトリのことを「親ディレクトリ」といいます。また，カレントディレクトリの中にあるディレクトリのことを「サブディレクトリ」といいます。ですから，BINディレクトリはルートディレクトリのサブディレクトリにあたります。逆にルートディレクトリはBINディレクトリの親ディレクトリです。

●カレントドライブを変更する

カレントディレクトリの変更方法を知ったついでに，カレントドライブの変更方法にも触れておきましょう。これは，

A>b:

のように，ドライブ名を入力するだけと簡単です。大文字でも小文字でもかまいません。これでカレントドライブがBドライブに変更され，プロンプトは，

B>

に変わります。そう，標準のプロンプトはカレントドライブ名を表示するようになっていたのです。

カレントディレクトリは，ドライブごとに保持されています。Aドライブのカレントディレクトリと Bドライブのカレントディレクトリは別々に設定しておくことができます。CDコマンドだけをつかうと，カレントドライブのカレントディレクトリ名が表示されます。ドライブ名を与えて，

A>cd b:

とすれば，Bドライブのカレントディレクトリ名を表示することができます。

●パス名と階層化ディレクトリ

FORMAT.Xの場所を表すのに，

A:¥BIN¥FORMAT.X

と書くことはお話ししました。「A:」をドライブ名，FORMAT.Xをファイル名と呼ぶのは前述のとおりです。ところで，ドライブ名とファイル名の間に書かれている文字列，

¥BIN¥

にも名前がついており，「パス名」と呼ばれています。パス（path）とは「通り道」という意味です。「¥BIN¥」は「（ルートディレクトリ）の中のBINディレクトリの中」と読めます。ルートディレクトリからFORMAT.Xに行くには，この通り道を通ればいいよと，パス名で指示しているわけです。

先ほどは「ドライブ名に続けて'¥'を入力するとルートディレクトリを指定したことになる」と説明しましたが，正しくは「パス名を'¥'で始めると，ルートディレクトリを指定したことになる」というのが正解です。逆にパス名を'¥'で始めなければ，カレントディレクトリを指定したことになります。

A>dir bin

は'¥'から始まっていませんので，「（カレントディレクトリの中の）BINディレクトリを表示する」という意味になります。この例ではドライブ名も省略されています。ドライブ名を省略するとカレントドライブを指定したことになりますので，より正確に読むなら，「（カレントドライブの，カレントディレクトリの中の）BINディレクトリを表示する」となります。

A>dir b:bin

なら，「Bドライブの（カレントディレクトリの中の）BINディレクトリ」です。

図4を見てください。これはシステムデ

ィスク内のディレクトリやファイルを模式的に表したものです。ルートディレクトリの下には（中には）COMMAND.XやSYSディレクトリ，BINディレクトリなどがあり，それぞれのディレクトリの中にはファイルが入っています。

もちろん，ルートディレクトリの中にBINディレクトリがあるように，BINディレクトリの中に別のディレクトリがあってもかまいません。ディレクトリの中にディレクトリがあり，そのまた中にディレクトリがある……というぐあいに，ディレクトリは階層構造にすることができます。これを一般に「階層化ディレクトリ」といいますが，このとき図4のようなイメージを頭に描くと理解しやすくなります。

あまりにこの図に親しみすぎて，「ひとつ上に戻って〜して」とか「いったん親に戻ってください」「ルート（ディレクトリ）の下の〜の下の〜にファイルが入っているからね」といった会話が交わされることも少なくありません。

●絶対パスと相対パス

パス名を指定する場合に，ルートディレクトリから順にディレクトリをたどる表記を絶対パス，あるいはフルパスといいます。これに対し，カレントディレクトリからたどっていく表記を相対パスといいます。

カレントディレクトリがBINディレクトリだとしましょう。図4のASK68K.SYSというファイルは，絶対パスなら，

¥SYS¥ASK68K.SYS

と表記できます。相対パスなら，

..¥SYS¥ASK68K.SYS

となります。こちらは，BINディレクトリからひとつ上に戻ってSYSディレクトリに降りた（早速使ってしまった）わけです。相対パスのメリットは，次のような場合に実感できるでしょう。

この原稿が，「原稿」ディレクトリの中の「Oh!X」ディレクトリの中の「1991年」ディレクトリの中の「6月号」ディレクトリに入っていると想像してみてください。カレントディレクトリをここまで移して執筆している途中で，先月号の原稿をもう一度読みたくなりました。あなたは絶対パスで指定しますか？

CDコマンドに慣れてくると，カレントディレクトリを移動するのではなく，自分がそのディレクトリに降りていくような錯覚に陥ることがよくあります。その結果，

「ねぇ，あのプログラムこのハードディスクに入ってるんだよねぇ。ないよ」

「いまどこにいるのさ」

などという会話が交わされることもしばしばです。

ファイルされたデータを見る

三種の神器の最後は，ファイルに入っているデータを画面に表示するコマンドです。これにはTYPEというコマンド名がついています。

A>type ¥bin¥ed.hlp

と，表示したいファイルのファイル名を与えて使います。

TYPEコマンドはどんなファイルでも画面に表示することができます。ただし，表示されたデータを人間が読めるかどうかは別問題です。

A>type ¥bin¥ed.x

としてみてください。目茶苦茶な文字列が表示されます。

ED.XはX68000の仕事ファイルです。このファイルにはX68000が直接実行できるマシン語が収められています。つまり，皆さんはマシン語を目で見たわけです。そんなものが読めるはずありませんね。

●拡張子の役割

読めるファイルもマシン語ファイルも，X68000は区別なく「ファイル」として扱います。そのため，ファイル名を見ただけで区別できるようにしようと，拡張子に意味が与えられています。

X	仕事を入れたマシン語ファイルです。読むことはできません
SYS	システム用の特別なファイルです。多くはマシン語ファイルで読むことはできません
BAT	システム用の簡単なプログラムを入れたファイルで，バッチファイルと呼ばれています。これは読むことができます
SWP	WP.Xの文書ファイルです。章だてなどの特別な情報が追加されていますが，とりあえず読むことはできます
DOC	ドキュメントを入れたファイルです。もちろん読むことができます。プログラムの説明などを収めるのによく使われます。ディスクの中にREAD ME.DOC（私を読んで）というファイルがある場合は，これをTYPEコマンドで表示するといいでしょう
TXT	ドキュメントを入れたファイルですが，改行が1段ごとにしか入っていない点がDOCファイルとは異なります。このため，画面を横一杯使った見にくい表示になります

主なところはこんなものでしょうか。このほかにも，アプリケーションが自分用のファイルであることを示すためにつける拡張子が，アプリケーションの数だけといってもいいくらい存在しています。

ここで挙げたもののうち，TYPEコマンドで読めるものについてはあとでもう一度取り上げることにします。

図4　階層化ディレクトリ

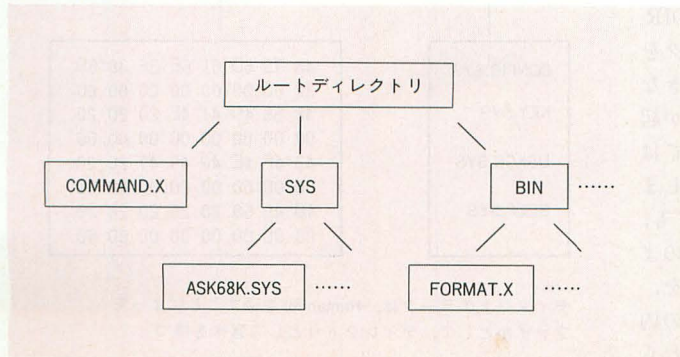



図5　TYPEコマンド

```
A>type ¥bin¥ed.hlp

        CTRLキー 機能一覧 1      (1/8)

CTRL+A   カーソルを1語後方（←）に移動
CTRL+B   カーソルを行の左端（または右端）に移動
CTRL+C   画面をロールアップ
CTRL+D   カーソルを1文字右に移動
CTRL+E   カーソルを1行上に移動
CTRL+F   カーソルを1語前方（→）に移動
CTRL+G   1文字削除
CTRL+H   バックスペース
CTRL+I   水平タブ
```

A＞からのアプローチ(2)

COMMANDマスターへの道

Izumi Daisuke
泉 大介

三種の神器を身につければA＞への恐れはもはやないはず。
次はCOMMAND.X上でひと通りのファイル操作を行うことが目標になる。
覚えたての知識をベースにいくつかの命令を使ってみよう。
また，Human68kとCOMMAND.Xの関係も理解しておきたい。

前項までの記事で，ディスクにどんなファイルが入っているのか調べること，カレントディレクトリを変更すること，そしてファイルを画面に表示することができるようになりました。お気づきかと思いますが，DIR，CD，TYPEというコマンドは，X68000の仕事ファイルとしてBINディレクトリに収められてはいません。これらのコマンドはいったいどこにあるのでしょうか。それと同時にもうひとつ気になっている点があろうかと思います。ルートディレクトリにあったCOMMAND.X。拡張子がXだということは，これはX68000の仕事を収めたファイルであるはずです。いったいどんな仕事を行うファイルなのでしょう。

Human68kとCOMMAND.X

ここまで，「システム」と曖昧な呼び方をしてきたものの正体をここではっきりさせておくことにしましょう。システムディスクを起動すると，ディスクからX68000を使う際の基本的なプログラムが読み込まれて動き始めます。ディスクドライブを「A:」「B:」などの名前で指定できるようにするのも，階層化ディレクトリを使えるようにするのも，すべてこのプログラムです。X68000単体ではファイルもディレクトリも扱えません。ディスクは単にデータがゴチャゴチャと入った磁気媒体という意味しか持たないのです(図1)。

X68000を操作（オペレーティング）する上で最も重要な役割を負っているこのプログラムを，「オペレーティングシステム(OS)」といいます。X68000のOSにはHuman68kという名前がつけられています。ディレクトリやファイルを扱えるようにすること以外にも，画面に文字を表示する機能，メモリの使用状況を管理して，複数のプログラムがメモリの同じ場所を異なった目的のために使うといったことがないように調整する作業など，Human68kはさまざまな仕事をこなしています。X68000の上で作業をする土台だとみなすことができるでしょう。47ページの図1で2行目に表示されていた情報「Human68k_Ver2」は，「このディスクはHuman68kの2番目のバージョンを収めたディスクである」という意味でディスクにつけられた名前です。

Human68kは土台を提供するもので，それ単体で何かできるというものではありません。キー入力を受け付け，それに応じた処理を行うといった，土台であるHuman68kと皆さんの間を取り持つプログラムが必要となります。この役目を負っているのがルートディレクトリにあったCOMMAND.Xです。COMMAND.X はプロンプトを表示し，入力されたコマンドをHuman68kの機能を使ってディスクから読み込み実行する役割を果たしています。システム起動時にHuman68kが読み込まれ土台が完成すると，Human68kはディスクからCOMMAND.Xを読み込み，作業を皆さんの手にゆだねるわけです。こうして皆さんがX68000を使う準備が整うことになります。

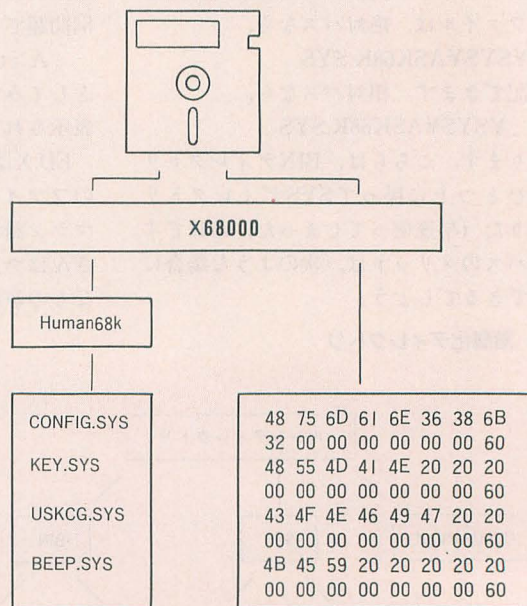
●外部コマンドと内部コマンド

ここまでに使ってきた三種の神器はいずれも基本的な仕事です。ディレクトリを見ようと思ってdirコマンドを入力したけど，DIR.Xの入っているディスクを抜いていたため実行できなかったなどという事態が起きては困ります(OS/9ではこのような事態が発生します)。いつでも，どこでも，実行できないと困るこのような基本的なコマンドを，COMMAND.Xは自分の内部に持っています。これらのコマンドは「内部コマンド」と呼ばれています。

FORMATやDISKCOPYなどの比較的重要度の低いコマンドはファイルにされています。すべての仕事をCOMMAND.Xが持っていると，COMMAND.Xのサイズがとんでもなく大きくなってしまうからです。内部コマンドと対比して，これらファイルにされたコマンドは「外部コマンド」と呼ばれます。

市販されているワープロや表集計，レイトレなどのプログラムは，拡張子がXの形式でファイルにされています。COMMAND.Xが外部コマンドを実行することができるようになっているおかげで，これらのアプリケーションプログラムをCOMMAND.Xから実行することもできるわけです。

COMMAND.XももちろんX68000の仕事ファイルですから，実行することができ

図1 Human68kとディスク

ディスク上のデータは，Human68kを通すことによって
ファイルとして，ディレクトリとして意味を持つ

ます。

　A>command

と入力してみてください。バージョンを表示したあと，今までと同じ画面になるはずです。画面は今までと同じですが，これは2つ目のCOMMAND.Xが動いている状態です。終了するには，

　A>exit

と入力します。EXITコマンドは内部コマンドで，COMMAND.Xを終了する命令です。続けてもう一度EXITコマンドを実行しようとすると，

　常駐しているので親プロセスに帰れません

と表示されます。難しいことをいってますが，これは「大もとのCOMMAND.Xだから終了できない」という意味です。最初に動いていたCOMMAND.Xがなくなっては，にっちもさっちもいかなくなってしまいますからね。

●コマンドマスターの秘訣

プロンプトに続けてコマンドを入力することで作業を進めていくこの世界では，コマンドを知っているかどうかが自由に行動できるかどうかの分かれ目になります。それぞれのコマンドの使い方まで覚え込む必要はありません。ただ，そのコマンドが「何をするためのコマンドなのか」という点だけを押さえておけばいいのです。使い方がわからなければマニュアルを見れば事足ります。また，外部コマンドにはコマンド名だけを入力すれば親切にガイドしてくれるものが多く含まれています。コマンド名だけの入力でガイドしてくれないものは，とりあえず上級者用のコマンドとして無視してもいいでしょう。BINディレクトリにはさまざまなコマンドが入っていますが，かなり長くX68000を使っている人でも一度も使ったことのないコマンドというのは存在するものです。

次の一歩を踏み出すために知っておいたほうがいいコマンドを，以下に取り上げることにします。内部コマンドはガイドが表示されませんので，基本的な使い方も併せて取り上げることにしましょう。

文書ファイルをまとめる

付属のワープロWP.Xで作成した文書ファイルを，データ保存用のディスクの「ワープロ」ディレクトリにまとめてみましょう。手順をざっと並べると，

1)　データ保存用ディスクのフォーマットを行う

2)　データ保存用ディスクに「ワープロ」ディレクトリを作る

3)　拡張子がSWPのファイルのコピーを作り，このディレクトリに収める

4)　元の文書ファイルが必要なければ削除する

となるでしょうか。順次詳しく見ていくことにしましょう。

●ディスクのフォーマットを行う

買ってきたばかりの生フロッピーはフォーマットしてから使うというのはすでにご存じでしょう。ほとんどの方が，ビジュアルシェルかSX-WINDOWでディスクのフォーマットを経験しているはずです。

Human68kがディスクにデータを記録する場合には，あらかじめ書き込まれたデータ記録位置を示すガイドを参照しながら行います。買ってきたばかりのディスクにはこのガイドが書き込まれていないため，データを保存することができません。そこでこのガイドをディスクに書き込む作業を「フォーマットする」とか「初期化する」と呼んでいるわけです。

コマンドモードではフォーマットは，FORMAT.Xという外部コマンドを使います。

　A>format

と入力すれば，画面にメニューが表示され，フロッピーディスクだけでなくハードディスクのフォーマットも選べるようになっています。

もしディスクに「データディスク」などの名前をつけたいなら，「ボリューム名」と書いてある項目を選択して名前をつければOKです。システムを転送するかどうかという項目は，フォーマットを行うディスクにHuman68kを入れるかどうかを設定します。データディスクとして使うなら，システムの転送は必要ありません。

また，フロッピーの場合なら，

　A>FORMAT B:

とするだけでもかまいません。ただし，ドライブ名を間違えないよう注意が必要です。フロッピーディスクから起動した場合は，ドライブ0がAドライブ，ドライブ1がBドライブになっていますが，ハードディスクから起動すると，ドライブ0がBドライブ以降になります。

すでに使っているディスクにワープロの文書をまとめる場合には，フォーマット作業は不要です。フォーマットを行うと，ディスクに記録されているデータはすべて破棄されますので注意してください。新たに買ってきたディスクを使う場合だけこの作業を行います。

●「ワープロ」ディレクトリを作る

次に，ワープロの文書を入れるためのディレクトリを作成しましょう。これは，

　A>md b:¥ワープロ

と入力すればOKです。

MDコマンドは内部コマンドで，Make Directory（ディレクトリ作成）を略したものです。ディレクトリはファイルを綴じておくものだと説明しましたが，使い方はまずディレクトリを作りその中にファイルを収めていくという手順になります。

●ディレクトリを削除する

ディレクトリを作成するコマンドを紹介したついでに，ディレクトリを削除するコマンドにも触れておきましょう。これにはRDコマンドを使います。

RDコマンドは内部コマンドで，Remove Directoryを略したものです。

　rd ディレクトリ名

の書式で使いますが，削除しようとするディレクトリにファイルが入っていたり，サブディレクトリがある場合には削除することができません。

●ファイルのコピーを作る

ファイルのコピーを作り，それを指定したディレクトリに収める。この作業を一気に行うのがcopyコマンドです。

　A>copy コピー元　コピー先

のように使います。

標準ではワープロはAドライブのQUICKSTARTディレクトリに文書ファイルを入れるようになっています。そこでコピー元は，

　A:¥QUICKSTART

となります。コピー先は，先ほど作ったディレクトリですから，

　B:¥ワープロ

です。リターンキーを押せばコピーが始まります。

このようにコピー対象をディレクトリ名にすると，そのディレクトリに入っているファイルがコピーの対象となります。注意が必要なのはコピー先にコピー元と同じファイル名のファイルが存在する場合です。このときコピー先にあるファイルは削除さ

れてしまいます。

コピー元，コピー先にはファイル名を指定することもできます。このとき，指定されたファイルのコピーが，コピー先として指定したファイル名で収められることになります。このときもコピー先にある同名のファイルは削除されます。

●便利なワイルドカード

ファイル名すべてを入力するのは面倒なものです。こんなときにはワイルドカードが便利です。ワイルドカードはファイル名本体や拡張子を補ってくれる特別な文字で，'*'で表現します。いくつか例を挙げましょう。

1)　*.SWP　：拡張子がSWPのもの
2)　特集1.*　：ファイル名本体が特集1であるもの
3)　特集*.*　：ファイル名本体が特集で始まるもの
4)　*.*　：なんでもOK

このようなワイルドカードをコピー元に指定すると，複数のファイルを一度に指定できます。ワープロの文書ファイルだけを対象にしたいなら，1)を利用すれば簡単です。ファイル名本体が1で終わるワープロのファイルを指定するなら「*1.swp」となるはずですが，このような指定はできないようになっています。できると便利なのですが。

その代わりといってはなんですが，任意の1文字を補うワイルドカードが用意されています。これは'?'で指定します。X68K_M.DICをコピーするなら，

　　?????m.dic

で指定できるので便利です。

●ファイルを削除する

コピーしたあと，元のファイルが必要なければ削除しましょう。ディスクの容量は限られていますからね。

これにはDELコマンドを使います。DELは削除を意味する単語deleteを縮めたものです。

　　del ファイル名

の書式で使います。

ディレクトリの中のファイルをすべて削除するには，

　　del ディレクトリ名

の書式で使います。このときは本当に削除していいかどうかを尋ねてきますので，OKならYキーを押してください。削除したファイルを復活させるコマンドはありませんので，注意が必要です。delコマンドでディレクトリ名を指定したり，ワイルドカードを使うときには，

　　A＞dir *.swp
　　A＞del *.swp

のように，一度dirコマンドで削除されるファイルを確認してから実行するといいでしょう。

システムのチューンアップ

Aドライブのシステムディスクを，ワープロのディスクに替えてみましょう。ワープロはWP.Xという名前でQUICKSTARTディレクトリに入っています。これまでと同じように，

　　A＞wp

と入力してみてください。ワープロが実行……できません。画面には，

　　コマンドまたはファイル名が違います

と表示されたはずです。

これは，基本的にCOMMAND.Xが実行できるのがカレントディレクトリにあるファイルに限られているからなのです。実行しようとするファイルがカレントディレクトリにない場合には，パス名を指定しなければなりません。この場合には，CDコマンドでQUICKSTARTディレクトリに移動するか，

　　A＞¥quickstart¥wp

と入力する必要があります。

「でも，FORMATはBINディレクトリにあるのに実行できたじゃないか」。お説ごもっとも。普段よく使うコマンドを実行するたびに，パス名をつけなければならないというのは面倒なものです。このためCOMMAND.Xに，カレントディレクトリにファイルがなければこのディレクトリを探しなさい，と指示しておくことができます。その指示とは「コマンド検索パス」と呼ばれ，pathコマンドで指定します。

pathコマンドは内部コマンドで，

　　A＞path a:¥;a:¥sys;a:¥bin;

のように，ファイルを検索したいパス名をセミコロン(;)で区切って指示します。パスの指定は，必ずフルパスで行ってください。単に，

　　A＞path

とだけ入力すると，現在設定されているコマンド検索パスが表示されます。「A:¥BIN」がコマンド検索パスとして設定されているのが確認できますね。BINディレクトリに収められたコマンドを実行できるのはこのためです。

Human68kのVer.2.0以降には，ヒストリと呼ばれる機能が付加されています。

　　A＞echo %path%

と入力して，画面にコマンド検索パスが表示されれば大丈夫。あなたのヒストリ機能は正常に動作しています。

このヒストリ機能があれば，

　　A＞path a:¥quickstart;%path%

と入力すれば，現在のコマンド検索パスの先頭に「a:¥quickstart」を補ってくれますので楽ちんです。また，

　　A＞path %path%

とタイプしておいて，ESCキー，'/'キーの順に押せばpathの後ろにコマンド検索パスがズラッと表示されます。あとはカーソル移動キーやBSキー，DELキーで自由に編集してリターンキーを押すだけで，コマンド検索パスを設定し直すことができます。

●QUICKSTARTを標準で設定する

ヒストリ機能によっていかに楽にコマンド検索パスを追加変更することができようと，毎回起動するたびに自分の手で設定するのはうんざりです。実際，BINディレクトリにパスが通っているといっても，そんな指示を与えた覚えはないでしょう。実はシステムディスクには，起動するときに自動的にこの指示を与えるように設定してあるのです。それがAUTOEXEC.BAT(オートエグゼックバット）というファイルです。

拡張子がBATのファイルをバッチファイルと呼ぶことは前の記事でお話ししました。バッチファイルは，テキストファイルです。typeコマンドで表示してみましょう。

　　A＞type autoexec.bat

と入力すれば，内容を見ることができます(図2-1)。ついでにワープロのディスクのAUTOEXEC.BATも表示してみましょう(図2-2)。

バッチファイルは，このようにCOMMAND.Xで入力するコマンドをズラリと書き並べただけのファイルです。

　　A＞autoexec

と入力すれば最初の行に書いたコマンドから順に実行されます。図2-1のシステムディスクのAUTOEXEC.BATは，コマンド検索パスを設定するだけです。実際に実行してみましょう。ここで設定されるコマンド検索パスは現在のコマンド検索パスと同

じですので，まず，

A>path a:¥

と入力してコマンド検索パスを変えてしまいます。

A>path

で変更されたことを確認してみてください。続いて，

A>autoexec

と入力します。ディスクが回ってAUTOEXEC.BATが実行されました。さて，コマンド検索パスは元に戻ったでしょうか。

このAUTOEXEC.BATというバッチファイルには，一連のコマンドを実行するバッチファイルとしての機能以外に，重要な役割があります。Human68kによってCOMMAND.Xが起動する際に自動的に行うことが，AUTOEXEC.BATという名前のバッチファイルの実行なのです。

ワープロのディスクをAドライブにセットして電源を入れると，自動的にワープロが実行されることをご存じだと思います。もう一度図2-2を見てみましょう。ワープロのディスクのAUTOEXEC.BATは，コマンド検索パスを設定したあとカレントディレクトリをQUICKSTARTディレクトリに移し，WP.X(ワープロ)を実行するように設定されています。電源を入れるとワープロが実行されたのはこのためです。

ここまでくれば，QUICKSTARTを標準でコマンド検索パスに設定する方法が見えてきたかと思います。そう，システムディスクのAUTOEXEC.BATを書き替えて，コマンド検索パスを設定しているところにQUICKSTARTを加えればいいのです。

●テキストファイルを書き替える

テキストファイルを変更するには，ED.Xを使うのが便利です。ED.Xはエディタ(編集するもの)と呼ばれ，テキストファイルを作成したり変更するためにBINディレクトリに用意されています。拡張子がXですから実行できるファイルで，

A>ed a:¥autoexec.bat

のように編集したいファイル名を与えて実行します。与えられたファイルが存在すればそのファイルの変更が行えますし，存在しなければ新たにファイルを作成することになります。

ED.Xでは，カーソルキーを使ってカーソルを移動し，自由に文字を書き込むことができます。またBS，DELキーも使えます。ただし，カーソルを移動できるのは，文字が書き込まれた最後の行までです。このほかにも，行頭にカーソルを移動したり，行末にカーソルを移動するなどの便利な機能が，CTRLキーを押したままでアルファベットキーを押すことで使えるようになっています。これらのキーの説明は，HELPキーを押せば表示されるようになっていますので，慣れるにしたがって少しずつ覚えていくといいでしょう。

コマンド検索パスにQUICKSTARTを追加するには，

path A:¥;A:¥SYS;……

の最初のAの上にカーソルを移動し，

path ■：¥;A:¥SYS;……

の状態から「a:¥quickstart;」とタイプすればOKです。画面は，

path a:¥quickstart;A:¥;……

となったはずです。リターンキーを押す必要はありません。リターンキーを押すと行が2つに分割されてしまいます。もし不幸にして行が分割されてしまったら，あわてずBSキーを押してください。これで元どおり上の行とくっつきます。

BSキーはカーソルの左の文字を削除する機能を持っています。画面に表示されてはいませんが各行の最後には「改行」という文字が書き込まれています。上の例はこの「改行」という文字を削除したわけです。ちなみに，ESCキーを押してからMキーを押すと，「改行」文字が画面に表示されるようになります。BSキーに対しDELキーは，カーソルの上にある文字を削除する機能を持っています。

AUTOEXEC.BATの変更が終了したら，ESCキー，Eキーを順に押すと，変更したファイルがディスクに書き込まれてED.Xが終了します。変更したファイルをディスクに書き込まず，強制的にED.Xを終了するにはESCキーを押してからQキーを押します。この2つの終了のしかたは覚えておきましょう。カーソル移動の方法，文字の削除方法，そしてED.Xの終了方法。これだけ覚えておけば，とりあえず十分です。

AUTOEXEC.BATの変更が終了したら，本当に自動的にコマンド検索パスがQUICKSTARTにも設定されるかどうか確かめてみましょう。リセットボタンを押してC，Human68kを再起動してください。

A>path

で表示してみましょう。大丈夫ですね。

メモリが2Mバイト以上になっているX68000なら問題はありません(コラム「デバイスドライバ」参照)。Aドライブのシステムディスクをワープロのディスクに入れ替え，

A>wp

と入力すればワープロが実行されます。

●RAMディスクを作る

RAMディスクはメモリの一部をあたかもフロッピーディスクのように使うものです。電源を切るとRAMディスク上のデータが消えてしまうという弱点はありますが，データの読み書きはハードディスクよりも高速ですので，十分使用するメリットはあります。ただでさえメモリのない1Mバイトのマシンにはお勧めできませんが，メモリが2Mバイト以上あるなら，ぜひとも試していただきたいものです。

RAMディスクを作るにはシステムディスクのSYSディレクトリに入っているRAMDISK.SYSを登録すればOKです。ED.Xを使って，CONFIG.SYSに，

DEVICE=¥SYS¥RAMDISK.SYS #M512

という1行を付け加えましょう。

A>ed config.sys

としてエディタを実行したら，「DEVICE=」がずらりと並んでいる場所にカーソルを移動させます。↓キーで移動できますね。

■EVICE = ¥SYS¥……

となりましたか？ では，リターンキーを押してください。これで，

■EVICE = ¥SYS¥……

のようにポッカリと1行空きました。↑キーでカーソルをこの空いた行に移動し，

DEVICE = ¥SYS¥RAMDISK……

とタイプすれば完了です。大文字でも小文字でもかまいません。「RAMDISK.SYS」のあとの「#M512」は，メインメモリの512KバイトをRAMディスクとして使いますという指示です。もっと小さくてかまわないなら，この数値を小さくしてください。

CONFIG.SYSの変更が終わったら，ESCキー，Eキーの順に押してエディタを終了します。そしてリセットです。変更したCONFIG.SYSは，システムを再起動して初めて有効になります。フロッピーディスクのA，Bドライブに加えて，RAMディスクがCドライブとして作成されたはずです。

このRAMディスクに，日本語辞書のX68K_S.DICをコピーしてみましょう。これはサブ辞書に相当するファイルです。Bドライブに辞書ディスクをセットして，

A＞copy b:¥x68k_s.dic c:
でコピーできます。

次にCTRLキーを押しながらXF1キーを押して日本語入力状態にしたら，ファンクションキーのF8を押してください。これは辞書のファイル名を指定する機能です。F8キーを押すたびに，メイン辞書，サブ辞書のファイル名を交互に尋ねてきます。サブ辞書のファイル名を，

C:¥X68K_S.DIC

と入力します。変更が終わったら，ESCキーを押すと終了です。

F9キーを押して辞書の学習情報をディスクに保存するように設定すると，そのぶん変換が遅くなりますが，サブ辞書をRAMディスクに入れてしまえばぐっと素早く変換してくれるようになります。これで思う存分学習させることができるでしょう。電源を切る前に，サブ辞書を辞書ディスクにコピーし戻すことを忘れてはいけませんが。

この設定が気に入ったら，最初からCドライブのサブ辞書を使うように変更してみましょう。これはCONFIG.SYSファイルの，

DEVICE=¥SYS¥ASK68K.SYS……

の行を変更します。行の中ほどに，

B:¥X68K_S.DIC

と書いてある場所がありますね。ここを

C:¥X68K_S.DIC

に書き替えてしまえばいいのです。

自分だけのシステムを作るために

CONFIG.SYSにこのように変更を加えることで，どんどん自分好みの操作環境は整ってきます。そのためにも，SYSディレクトリに入っているさまざまなデバイスドライバが何をするデバイスドライバなのかを知っておくことは重要だといえるでしょう。コマンド名と同じです。マニュアルを隅から隅まで暗記する必要はありませんが，自分がしたいと思ったことが実現できるのかどうか，また，実現するにはどうすればいいのかを，マニュアルから探せる程度にはなっておきたいものです。

長々と説明してきたことは，SX-WINDOWなどを使えばより直感的に行えるものが多いのですが，コマンドモードを使い込めば，もっと柔軟で複雑な処理を行うことも可能になります。また，システムの基本的な概念やちょっと立ち入ったHuman68kの動作を理解するのにも役立つでしょう。コマンドモードの突っ込んだ使い方に興味のある方は，マニュアルの「リダイレクト」「パイプ」や「ヒストリドライバ」のページを覗いてみてください。

図2 AUTOEXEC.BATの内容

```
1)  システムディスクのAUTOEXEC.BAT

PATH A:¥;A:¥SYS;A:¥BIN;A:¥BASIC2;A:¥ETC

2)  ワープロディスクのAUTOEXEC.BAT

PATH A:¥;A:¥SYS;A:¥QUICKSTART;
CD ¥QUICKSTART
WP
```

図3 CONFIG.SYSファイルの内容

```
1)  ワープロディスクのCONFIG.SYS

A>type config.sys
FILES      = 15
BUFFERS    = 20 1024
LASTDRIVE  = Z:
KEY        = ¥KEY.SYS
USKCG      = ¥USKCG.SYS
BELL       = ¥BEEP.SYS
DEVICE     = ¥SYS¥SCSIDRV.SYS /ID0
DEVICE     = ¥SYS¥PRNDRV.SYS
DEVICE     = ¥SYS¥PCMDRV.SYS
DEVICE     = ¥SYS¥ASK68K.SYS B:¥X68K_M.DIC B
:¥X68K_S.DIC ¥ASK¥ENV1.ASK
DEVICE     = ¥SYS¥FLOAT2.X
DEVICE     = ¥SYS¥HISTORY.X /D¥HIS¥ /SH2,8,4
ENVSET     = 512 ¥WP.ENV
VERIFY     = ON

2)  システムディスクのCONFIG.SYS（一部）

DEVICE     = ¥SYS¥SCSIDRV.SYS /ID0
DEVICE     = ¥SYS¥PRNDRV.SYS
DEVICE     = ¥SYS¥PCMDRV.SYS
DEVICE     = ¥SYS¥ASK68K.SYS B:¥X68K_M.DIC B
:¥X68K_S.DIC ¥ASK¥ENV1.ASK
DEVICE     = ¥SYS¥RSDRV.SYS
DEVICE     = ¥SYS¥OPMDRV.X
DEVICE     = ¥SYS¥FLOAT2.X
DEVICE     = ¥SYS¥HISTORY.X /D¥HIS¥ /SH2,8,4
DEVICE     = ¥SYS¥IOCS.X
```

デバイスドライバ

最新のシステムディスクから起動した場合，COMMAND.Xから起動し，ちゃんとパスが通っていても，メモリが1Mバイトしかなければ，

A＞wp

としても残念ながらワープロは実行できません。ワープロが動くのに十分なメモリが残っていないからです。ワープロのディスクを起動したときには動くのに，システムディスクを起動したときには動かない。この違いはどこにあるのでしょうか。

●COMMAND.Xが動くまで

Human68kがディスクをA，Bなどの名前で扱えるようにしたり，ファイルやディレクトリを扱えるようにするための土台を作るものであることはお話ししました。このほかにも画面やプリンタに文字を出力できるようにしたり，RS-232Cを通信に使えるようにしたりと，Human68kの土台作りは多岐に及んでいます。

これらの土台のすべてをHuman68kが自分の内部に用意しているわけではありません。すべてを自分の内部に持っていたのでは，システムが大きくなりすぎますし，新しい周辺装置が出てきたときにHuman68k自身に変更を加えなければならなくなってしまいます。そこで画面やキーボード用の土台，プリンタ用の土台，RS-232C用の土台，と土台を機能ごとに分割し，これらを取捨選択して土台作りを進めていくようになっています。

それぞれに分割された土台はデバイスドライバと呼ばれます。周辺装置（デバイス）を制御するもの（ドライバ）というわけです。デバイスドライバには通常SYSという拡張子が与えられ，SYSディレクトリの中にまとめられています。これらデバイスドライバは，画面やキーボード，フロッピーディスクドライブ，ハードディスクといった基本的なものを除いて，ユーザーが使用するかどうかを選択できるようになっています。

図3-1をご覧ください。これはワープロのディスクに入っているCONFIG.SYSというファイルです。CONFIG.SYSは拡張子がSYSですがテキストファイルで，このように表示することができます。Human68kは，このファイルを参照しながら，自分の動作環境を整えます。最初の数行は特に意識する必要はありません。注目してほしいのは，

DEVICE =

で始まる行です。この行がデバイスドライバを「使うよ」と宣言している部分です。最初はSYSディレクトリの中のSCSIDRV.SYSというデバイスドライバを使うよと指示しています。これはSCSI装置を制御します。その次はPRNDRV.SYSで，これはプリンタを制御するデバイスドライバです。必要なデバイスドライバは，すべてこのようにCONFIG.SYSに書き込んでおくことによって使えるようになるのです。デバイスドライバを使えるようにするこの作業を，「デバイスドライバを登録する」あるいは「デバイスドライバを組み込む」といいます。

図3-2はシステムディスクのCONFIG.SYSのデバイスドライバを登録している部分です。ワープロディスクのCONFIG.SYSより多くのデバイスドライバが登録されているのがわかりますね。

デバイスドライバといえどプログラムです。それを組み込むのにはメモリが必要です。1Mバイトしかメモリのない機種でもワープロを実行できるように，ワープロのシステムディスクではとりあえず不必要なデバイスドライバを組み込まないことによってメモリの空きを確保しようとしているわけです。

メインメモリが2Mバイト以上でハードディスクを使っているなら，ワープロもハードディスクに収めて快適に使いたいと思うのは当然です。が，ワープロのディスクに入っているCONFIG.SYSをハードディスクに入れてしまうと，OPMドライバが登録されませんので音楽を演奏できなくなってしまいます。RS-232Cも使えません。システムをハードディスクに組み込んだら，ワープロのディスクからはQUICKSTARTの中身だけ入れておけばよいでしょう。

上級者のための環境考

Oh!X編集部のマシンルームには，新旧取り混ぜ多くのX68000が稼動しています。スタッフや寄稿ライターはこれらのマシンを使って，購入できないMOに感動したり，（本来自宅で書いてくるはずの）原稿を執筆したりしています。

ただ，これだけのマシンが混在すると1人で1台のマシンを使っていたときには予想もつかない事態が発生することになります。その最たるものは，

フロッピーディスクドライブが
どこだかわかんな〜い！

Human68kは，システムが起動されたメディアから順にA，B……と名前をつけていきますので，ハードディスクのパーティションの数により，さらには光磁気ディスクのパーティションの数により，フロッピーディスクドライブにつけられる名前は異なってきます。そこで提案。

フロッピーディスクドライブは
A，Bに固定しよう！

これなら，どんなドライブ構成のマシンの前に座っても迷うことはありません。

driveコマンドは，

drive

と入力すればAドライブから順に，それぞれのドライブがどのメディアにあたっているのかを表示してくれるコマンドです。このコマンドには，

drive a: c:

のようにドライブ名を2つパラメータに指定する使い方があります。これは，それまでのAドライブをCドライブに，それまでのCドライブをAドライブに入れ替えなさいという指示です。この機能を使ってフロッピーディスクドライブを常にA，Bドライブに振っておけばいいわけです。

ただ，これには若干問題があります。仮にハードディスクがつながったX68000があり，ハードディスクには4つのパーティションがあるとしましょう。ハードディスクからシステムを起動すると，A〜Dがハードディスク，E，Fがフロッピーディスクになります。ここで，

drive a: e:

drive b: f:

としてドライブを入れ替えると，システムを起動したドライブがEドライブとなり，シス

リスト1

```
  1: /*********************************************
  2:      ドライブ名を付け替えるコマンド
  3:  *********************************************/
  4:
  5: #include <stdio.h>
  6: #include <stdlib.h>
  7: #include <fctype.h>
  8: #include <doslib.h>
  9:
 10: void setNewName( int drvid, char *param );
 11: int  dname( int );
 12: void help( void );
 13: void error( int mode, int d );
 14:
 15: struct DPBPTR   dp;
 16: char newDrive[ 27 ];
 17: char usedDrive[ 27 ];
 18: char ramdisk[] = "A:";
 19: int  chgflag = 0;
 20:
 21: void main( int argc, char *argv[] )
 22: {
 23:     int     i, j, tmp;
 24:     int     nameptr, newdrv;
 25:     int     check = 1, ch;
 26:     char    id[ 3 ];
 27:
 28:     if ( argc > 1 )
 29:         /* パラメータセット */
 30:         for ( j=1; j<argc; j++ )
 31:             if ( argv[ j ][ 0 ] == '-'  || argv[ j ][ 0 ] == '/' )
 32:                 /* スイッチに応じて新しいドライブ名をセット */
 33:                 switch ( toupper( argv[ j ][ 1 ] )) {
 34:                   case 'F':
 35:                     setNewName( 0xFE, argv[ j ]+2 );
 36:                     break;
 37:                   case 'R':
 38:                     setNewName( 0xF9, argv[ j ]+2 );
 39:                     break;
 40:                   case 'N':
 41:                     check = 0;
 42:                     break;
 43:                   case '#':          /* -#f7z ID=F7のドライブをZ:にする */
 44:                     strncpy( id, argv[ j ]+2, 2 );
 45:                     setNewName( strtol( id, NULL, 16 ), argv[ j ]+4 );
 46:                     break;
 47:                   default:
 48:                     help();
 49:                     break;
 50:                 }
 51:             else {
 52:                 /* ドライブ名並びに応じてドライブ名をセット */
 53:                 nameptr = 0;
 54:                 for ( i=1; i<=26; i++ ) {
 55:                     if ( newDrive[ i ] != 0 ) continue;
 56:                     if ( argv[ j ][ nameptr ] == 0 ) break;
 57:                     newDrive[ i ] = dname( argv[ j ][ nameptr ] );
 58:                     if ( usedDrive[ newDrive[ i ]] != 0 )
 59:                         error( 1, usedDrive[ newDrive[ i ]] );
 60:                     usedDrive[ newDrive[ i ]] = 1;
 61:                     nameptr++;
 62:                     chgflag = 1;
 63:                 }
 64:             }
 65:
 66:     /* ドライブ名を指定されなかったドライブに名前をセット */
 67:     for ( i=1, nameptr=1; i<=26; i++ )
 68:         if ( newDrive[ i ] == 0 ) {
 69:             while ( usedDrive[ nameptr ] ) nameptr++;
 70:             newDrive[ i ] = nameptr++;
 71:         }
 72:
 73:     if ( check ) {
 74:         /* チェックモードなら新旧対応表を表示 */
 75:         for ( i=1; i<=26; i++ ) {
 76:             printf( "%c:", i-1+'A' );
 77:             if ( GETDPB( i, &dp ) < 0 )
 78:                 if ( CHGDRV( i-1 ) == i-1 )
 79:                     printf( "仮想ドライブ" );
 80:                 else
 81:                     printf( "装置情報なし" );
 82:             else
 83:                 switch ( dp.id ) {
 84:                   case 0xFE:
 85:                     printf( "FD %d¥t", dp.unit );
 86:                     break;
 87:                   case 0xF9:
 88:                     printf( "RAMDISK %d", dp.unit );
 89:                     break;
 90:                   default:
 91:                     printf( "ID( %02X )", dp.id );
 92:                     break;
 93:                 }
 94:             if ( chgflag )
 95:                 printf( "¥t→ %c", 'A'-1+newDrive[ i ] );
 96:             printf( "¥n" );
 97:         }
 98:         if ( chgflag ) {
 99:             /* ドライブ名が変更されていたら、変更するか確認する */
100:             printf( "¥n以上のように変更します (Y/N) " );
101:             ch = getch();
102:             printf( "¥n" );
103:         } else
104:             ch = 'N';
```

テムと一緒に入れてあることの多いHuman68kのコマンドなどもＥドライブにいってしまいます。そういうものだと割り切れればいいのですが，個人的にはどうも好きになれない。Human68kのコマンドは，フロッピーディスクの次のドライブＣにいてほしいのです。

事態はここに至って俄然複雑さを増してきます。２つのドライブを入れ替えながら

HD0　HD1　HD2　HD3　FD0　FD1

↓

FD0　FD1　HD0　HD1　HD2　HD3

と並び替えるための手順を考え出さなければなりません。この例では結局driveコマンドを４回使って並び替えることになりますが，これに光磁気ディスクでも加わろうものなら地獄のような事態が発生します。

そこで用意したのがリスト１です。これはドライブの入れ替えではなく，ドライブの並べ替えを行なうコマンドのプログラムリストです。使い方は簡単で，現在のＡドライブから順に，新しく与えたいドライブ名を書き並べていくだけです。上の例なら，

rendrv cdefab

と入力すればOKですし，フロッピーディスクをＡ，Ｂにするだけなら，

rendrv /fab

のようにfオプションを使うこともできます。同様にRAMディスクのドライブ名を指定するにはrオプションを使用します。いずれも並べ替えの前にドライブ一覧を表示して確認を求めてきます。ｎオプションは，この確認画面を表示しないようにする指示です。

プログラムは読みやすさを考慮してメッセージ表示をprintf関数で行っています。このままでも使えますがrendrv.xのサイズが気になる方はメッセージ表示にB_PRINTを使うなどの変更を加えてください。

rendrvコマンドのもうひとつの機能は，以前のOS特集のときに囲みで紹介された，drvコマンドの機能を含んでいることです。drvコマンドは，RAMディスクのドライブ名を環境変数ramdiskにセットするコマンドです。rendrvコマンドを実行すると，ドライブの並べ替えのあと自動的にRAMディスクのドライブ名を環境変数raMdiskに設定します（Ａから順にたどっていき，最初に見つけたものを設定する）。

すべてのマシンはフロッピーディスクをＡ，Ｂドライブに，そしてRAMディスクを512Ｋバイト以上確保しているマシンは起動後のカレントドライブをRAMディスクに。これこそ，清く正しい上級ユーザーの姿勢だと思うのですがいかがでしょうか。

```
105:        } else if ( chgflag )
106:            /* チェックモードでなく、ドライブ名が変更されている場合 */
107:            ch = 'Y';
108:        else
109:            /* チェックモードでなく、ドライブ名が変更されていない場合 */
110:            ch = 'N';
111:
112:        switch ( toupper( ch )) {
113:          case 'Y':
114:          case 13:  /* CR */
115:            newdrv = newDrive[ CURDRV()+1 ] - 1; /* 新カレントドライブ */
116:            for ( i=1; i<26; i++ ) {
117:                for ( j=i; j<=26; j++ )
118:                    if ( newDrive[ j ] == i ) {
119:                        DRVXCHG( i, j );
120:                        tmp = newDrive[ i ];
121:                        newDrive[ i ] = newDrive[ j ];
122:                        newDrive[ j ] = tmp;
123:                        break;
124:                    }
125:            }
126:            CHGDRV( newdrv ); /* 新カレントドライブをセット */
127:            printf( "ドライブを変更しました¥n" );
128:            break;
129:        }
130:
131:        /* 環境変数 ramdisk をセット */
132:        for ( i=1; i<=26; i++ )
133:            if ( GETDPB( i, &dp ) < 0 )
134:                continue;
135:            else if ( dp.id == 0xF9 ) {
136:                ramdisk[ 0 ] = 'A'-1+i;
137:                SETENV( (UBYTE *)"ramdisk", NULL, (UBYTE *)ramdisk );
138:                break;
139:            }
140: }
141:
142: /*
143:  *      スイッチに応じて新しいドライブ名をセット
144:  */
145: void setNewName( int drvid, char *param )
146: {
147:        int     i, nameptr;
148:
149:        nameptr = 0;
150:        for ( i=1; i<=26; i++ ) {
151:            if ( GETDPB( i, &dp ) < 0 ) continue;
152:            if ( dp.id == drvid ) {
153:                if ( param[ nameptr ] == 0 ) break;
154:                if ( newDrive[ i ] != 0 )
155:                    error( 0, i );
156:                newDrive[ i ] = dname( param[ nameptr ] );
157:                if ( usedDrive[ newDrive[ i ]] != 0 )
158:                    error( 1, usedDrive[ newDrive[ i ]] );
159:                usedDrive[ newDrive[ i ]] = 1;
160:                nameptr++;
161:            }
162:            chgflag = 1;
163:        }
164: }
165:
166: /*
167:  *      ドライブ名が正当かどうかをチェック
168:  */
169: int dname( int name )
170: {
171:        name = toupper( name ) - 'A' + 1;
172:        if ( name < 1 || 26 < name )
173:            help();
174:        return( name );
175: }
176:
177: /*
178:  *      ヘルプを表示して作業中止
179:  */
180: void help( void )
181: {
182:        printf( "ドライブ名を変更します¥n" );
183:        printf( "使用法:rendrv  [(-・/)スイッチ] [ドライブ名] [(-・/)スイッチ]¥n" );
184:        printf( "スイッチ  fドライブ名  FDに設定するドライブ名¥n" );
185:        printf( "          rドライブ名  RAMディスクに設定するドライブ名¥n" );
186:        printf( "          n            ドライブ設定の確認なし¥n" );
187:        printf( "ドライブ名を最初に指定すると、現在のAドライブから¥n" );
188:        printf( "順に指定されたドライブ名をセットします。¥n" );
189:        printf( "スイッチのあとにドライブ名を指定すると、Aドライブ¥n" );
190:        printf( "から順にまだドライブ名のついていないドライブを探し、¥n" );
191:        printf( "指定された名前をセットします¥n" );
192:        printf( "ドライブ名が実際のドライブ数より少ない場合は、足り¥n" );
193:        printf( "ない部分を指定されていないドライブ名で補います¥n" );
194:        printf( "使用例:rendrv -fab -re → FDをA:B:に、RAMDISKをE:にします¥n" );
195:        printf( "        rendrv cab      → A:B:C:をC:A:B:にします¥n¥n" );
196:        exit( 1 );
197: }
198:
199: /*
200:  *      エラーを表示して作業中止
201:  */
202: void error( int mode, int d )
203: {
204:        if ( mode )
205:            printf( "%C: を重複して指定しました", 'A'-1+d );
206:        else
207:            printf( "%C: は既に %C: に変更されています",'A'-1+d,'A'-1+newDrive[ d ] );
208:        printf( "¥n¥x07" );
209:        exit( 1 );
210: }
```

〔泉氏の意見に対する編集室の反応〕違うと思う（MU)。SUPERならＰとＱ，XVIはＥとＦ。覚えればすみます（U)。ウィンドウだからどこでも関係ないもん（S.S.)。決めてかかるとかえって危険でしょ？（S.N.)

貴方はどのタイプ?

SX-WINDOWで環境をつくること

Yoshida Kouichi
吉田 幸一

今後のX68000の環境を占う最も重要なユーザーインタフェイス。
それがSX-WINDOWだ。新機種の発表とともにバージョンもVer1.10となり,
これだけでもある程度便利に使える機能を持つに至った。
ここでは,SX-WINDOWで環境をつくるということを考えてみよう。

男と女の間に暗くて深い川があるように,パソコンと人間の間にも暗くて深い川がある(っても,『黒の舟歌』なんて誰も知らないだろうな)。暗くて深い川の渡し舟。その船頭がOSであり,船頭の乗る船がシェルである。

なんのこっちゃ。まあ,渡し舟に乗って此岸と彼岸を行ったり来たりするのがなかなかオツなものだ。その渡し舟は電子の速度で走る高速船だ。ときどき,彼岸へいったきり戻らないやつもいたりして,それもまた人生である。

船を無理矢理シェルとこじつけたからには,それなりに言い訳せねばならないわけで,小さくて軽い木の船がコマンドシェルだとすれば,豪華客船まではいかなくとも,加山雄三のクルーザーがウィンドウシステムだ。ひでえ喩えだな。

そういうわけで,軽くて小回りのきくボートと贅沢にパイプをくゆらせながら走るクルーザーである。誰が川を渡るだけなのにクルーザーに乗るんだ,という話もあるが,そこはそれ,人間は贅沢になるものなのだ。黄河くらい広い川だと思ってもらってもいい。

そういうわけで,SX-WINDOWに乗って暗くて深い川を彼岸へと渡ろう。

*

SX-WINDOWというのは何かというと,図を見てもらえればわかる。ポイントは2つだ。COMMAND.Xより守備範囲が広くて,COMMAND.Xよりでかい。だから,そういう贅沢な環境を使おうと思ったら要2Mバイトで要ハードディスクである。ハードディスクはなくても使えるが,マウスオペレーティングの快適さと,ディスク入れ替えガシガシは似合わない。

さて,SX-WINDOWの使い方には図1.bと図1.cの2通りがある。見てわかるとおり,COMMAND.X上から立ち上げる方法といきなり,SXWIN.Xを立ち上げる方法だ。

先に,SX-WINDOWを使うに必要な境というものを見ていこう。

SX-WINDOWは何を求めるか

SX-WINDOWを実行するためには,Human68kの立ち上げ時に次のデバイスドライバが必要だ。

FLOAT2.X ないしはそれに準ずるもの。これがないと起動しないプログラムはたくさんある。入れておくべし。

FSX.X 実のところ,SX-WINDOWのさまざまな処理はSXWIN.Xではなく,このFSX.Xが常駐して行っているのだ。

FSX.Xは非常に大きいので,いつも常駐させておくには荷が重い。

しかし,FSX.XはCONFIG.SYSに書かなくともCOMMAND.X上から常駐させたり,解除させたりできるのだ。図1.bなら,

```
fsx
sxwin
fsx－r
```

というバッチファイルを作っておけばいい。立ち上げ時に200Kバイト以上あるFSX.Xの起動が加わるからちょっと時間がかかるけどね。それからウィンドウを終了(システムアイコンで終了を選択)したら,FSX.Xを解除するようにしたい。そこで最後の－rに注目。以前村田氏が,解除のためのスイッチがないことを指摘していたが,これはVer.1.10からついたスイッチだ。

上の2つに加えて,OPMしたい人はOPMDRV.Xが必要なのはいうまでもない。日本語処理したい人は,ASKかFIXER4を組み込んでおく必要がある。そういうもののことも考えておく必要があるのだ。

ちなみに,COMMAND.X上からFSX:XSを組み込んでSXWINするのと,CONFIG.SYSに組み込んでしまうのと比べると,フリーエリアの差は約35Kバイト(独自調査による)しかなかった。SX-WINDOW専門で突き進むのでなければ,COMMAND.Xからマニュアル起動もいいだろう。

ここで,図1.bの人をタイプB,図1.cの人をタイプCと呼ぼう。タイプAはSX-WINDOWを使わない派である。でも,ハードウェア(メモリとハードディスク)に余裕があるなら,複数ファイルのコピーや複数ファイルの参照・編集など,SX-WINDOWがおいしいシーンもあるので,タイプAの人もときどきは使ってあげよう。

図1 command.xとSX-Windowの概念図

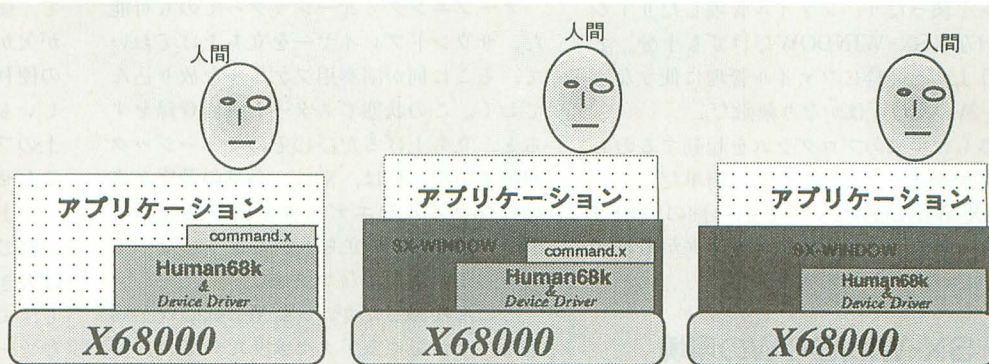



図1.a command.xの場合
図1.b command.x + SX-Window の場合
図1.c SX-Window のみの場合

図2

吉田幸一のデスクトップ

タイプBの環境整備

私はどうもこのタイプBを勧めるのではないかと思われるふしがあるかもしれないが，あにはからんや。

タイプBはコマンドシェル環境をメインにしながら，ときどきSX-WINDOWも楽しもうという贅沢ものである。だから，コマンドシェル上での環境整備がどうしても中心となる。HISTORY.Xを組み込んだり，RAMディスクを確保したり，IOCSを組み込んだり，とにかく，コマンドシェル上の環境を重視する。すると，SX-WINDOW実行時のフリーエリアがいくらか少なくなってしまう。いまのところSX-WINDOW上の大きなアプリケーションはないのでそれでもいいが，将来，200Kバイトクラスのアプリケーションが出てきたときとか，小さなウィンドウをたくさん開いて作業するときにちょっと怖い。

どっちにしても，コマンドシェルの知識が必要となるので，へらへらとX68000を使いたい人にはあまり向かない。

最初からSX-WINDOW

そしてタイプC。たとえば，EXPERTII，PROII以降の機種を持っていて，内蔵もしくは外付けのハードディスクをつないでいる比較的エンドなユーザーに多いだろう。「ディスクのフォーマットって，なに?」という初心者ユーザーにとって，ディスクを入れると自動的にフォーマットしてくれるSX-WINDOWは重宝する。

まだまだ，COMMAND.Xの世話になったほうがよいこともあるが，アプリケーションを使ったり，ファイル管理したりするだけならSX-WINDOWだけでも十分。というより，純粋にファイル管理に使うならSX-WINDOWはかなり無敵だ。

さらに従来のプログラムを起動するのもダブルクリック一発だから，簡単だ。

SX-WINDOWといっても，今回のVer.1.1なら，エディタがある。エディタがあれば，たいていのことはできるのだ。

SX-WINDOWの環境

SX-WINDOWを手に入れたら，2つのことをしなければならない。

ひとつはコントロールパネルとスイッチである。どちらもX68000のシステムアイコンのポップアップメニュー内にある。ここで各種ハードウェアの設定やらSX-WINDOW上での環境設定をするわけだ。時刻合わせやプリンタ設定などなどである。

これが終われば，画面設計だ。各種アイコンの位置や，背景やら，スタート画面設定やらとあれこれ遊べるものが多い。

たとえば，私のはこうなっている。ずいぶんデフォルトとは変えてしまった。

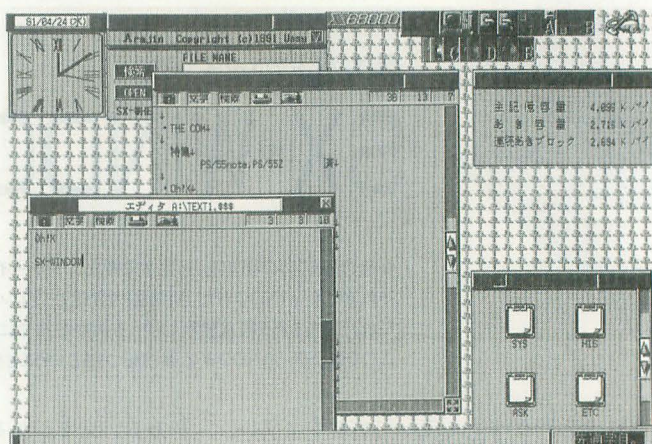
え? システムアイコン? ふふふ。OPT.1を押しながら左クリックすれば持ち上げることができるのだ。そして，ドラッグして適当な位置に置く。ほうらできた。もっとも，Ver.1.1で新しく追加された機能だから，旧バージョンの人は新バージョンが発売され次第，アップデートするように。

それから，ドライブアイコン。ドライブトレイから必要なものだけをセレクトして出しておく。無計画なドライブ構成をしていると，私みたいにごちゃごちゃとドライブアイコンが並ぶことになる。

続いて，レイアウト。どこにシステムアイコンを置き，どこに時計を置き，何を常駐させておくか。いつも同じ画面からスタートするか，終了時の画面を登録するか。背景はどうするか。などなどである。このレイアウトは意外と重要だ。時計なんてのはいつも出しておきたいものだ。

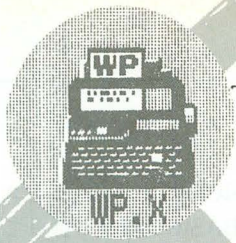
オープニングミュージックってのも可能だ。サウンドプレイヤーを立ち上げておいて，そこに何か演奏用ファイルを放り込んでおく。この状態でスタート画面登録をすると，立ち上げるたびにそのミュージックが鳴るのだ。私は，常に，今月の予定を書いたファイルをエディタで読み込む形でSX-WINDOWが立ち上がるようになっている。これはなかなか快適だ。

とこんな感じだが，SX-WINDOWを購入してついてくるアクセサリだけでは，特筆すべき環境は作れない。実のところ，先月の付録ディスクにちょっとお世話になる。

特に，プログラムトレイとSXWHEREはぜひとも画面に常駐させておきたい。あまりにも便利だからだ。今月号でも116ページに詳しい解説があるので参照されたい。

もともとウィンドウシステムというのは，コマンドシェルでいうパスというものがない。ファイルやディレクトリの数が増えると，目的のファイルに辿り着くのにとても苦労するようになる。それを防ぐのがSXWHEREだ。

さらに，SX-WINDOWはファイルをダブルクリックすることによってそのファイルが起動したり，アイコンに登録されたプログラムが起動したりする。そういうとき，上記と同じ理由によって，いちいち目的のファイルを探してからダブルクリックでは，ウィンドウシステムが売りにしている機動性が損なわれてしまう。それを防ぐのが，プログラムトレイだ。登録されたプログラムなら，すぐ起動できる。もっとも，SX-WINDOW上で動くプログラムだけなので，コマンドシェル上で動くプログラムに関しては使えないようだ。

ウィンドウシステムの常として，発足したばかりではなかなか環境が整わない。Macintoshだって，MS-Windows 3.0だって，使いやすくするためにはフリーウェアが欠かせない。また，コマンドシェル環境の便利なプログラムには使いこなすのが難しいものもあったが，ウィンドウシステム上のフリーウェアはたいてい誰にでも使いこなせるインタフェイスを持っている。

つまり，ウィンドウシステムはユーザーが育てるものだ。メーカーやソフトハウスは大きなアプリケーションは出してくれても，より使いやすい環境整備のための小さなツールは採算を取るのが難しいためか，なかなか市販されないのが現状なのだ。そういう意味では，まだまだSX-WINDOWは育ち続けるのである。

ワープロからエディタへ

基本はテキストファイル

Saitou Susumu
斎藤　晋

**パソコンで自分なりの環境をつくれるために必要なのこと。
それは，簡単なファイル操作とテキストファイルの編集です。
日本語ワープロしかわからないという人，
ぜひともテキストファイルの扱いを覚えましょう。**

コンピュータを買って間もない初心者にとって，「自分なりの環境を」なんてことをいわれても，実感として理解するのは難しいかもしれない。そりゃあ，メモリやハードディスクの容量は大きいに越したことはないし，便利なソフトはあったほうがいい。そういうことなら，誰にだってわかるだろう。でも，環境というのはそれらの大切な資源が使用状況に反映されてこそのものなのだ。

残念ながら，こうこうこういう使い方が正しい，といったアドバイスはできない。パソコンには人それぞれに合った使い方というものがあるからね。それは結局自分で見つけていくしかないものだ。でも，なんとなく便利に使ってみたいという初心者の皆さんには参考になることもいくつかあるものだ。

パソコンの仕事のなかで，多くの人に共通して便利な機能は主として文字情報を扱うことだろう。べつにビジネスに利用するのでなくても，アドレス帳ぐらいは持っているだろうし，連絡事項をメモすることぐらいはあるだろう。学生だってレポートぐらいは書くだろう。実際には手書きで書かなきゃならないことでも，パソコン上で下書きを書くというのは有効だ。

作文が大の苦手だった僕でもなぜかこうしてOh!Xの原稿を書いている。はっきりいって今じゃキーボードに向かわないとなんだか考えがまとまらないくらいなのだ。

テキストファイルと文書ファイル

で，コンピュータで扱う「文字情報」の単位は主として「ファイル」という形になる。ワープロの文書ファイルなどがその代表例だ。

そこで本題に入るが，文字情報を扱ったファイルにはおおよそ2つのレベルがある。ひとつは「テキストファイル」と呼ばれているもの，もうひとつは「文書ファイル」などと呼ばれているものだ。

●テキストファイル

テキストとは文字列のこと。アルファベットや記号，かな，漢字などのそれぞれの文字には1バイトないし2バイトのコードが与えられている。それらの文字コードを並べてファイル化したものがテキストファイルだ。テキストファイルは純粋に文字情報だけを扱ったものと考えればよい。改行や文字列の終わりを示すコードはあるが，書式に関する情報は一切入っていない。「ただのテキスト」とか「べたテキスト」とかいろんな言い方をする。

テキストファイルは味もそっけもない文字列しか扱わないが，それだけに機種やアプリケーションの違いを超えて利用できるメリットがある。X68000のOSであるHuman68kはMS-DOSと互換性のあるファイルフォーマットを採用しており，テキストファイルはPC-9801やJ-3100などのDOSマシンとのあいだで双方向に受け渡しが可能となっている。

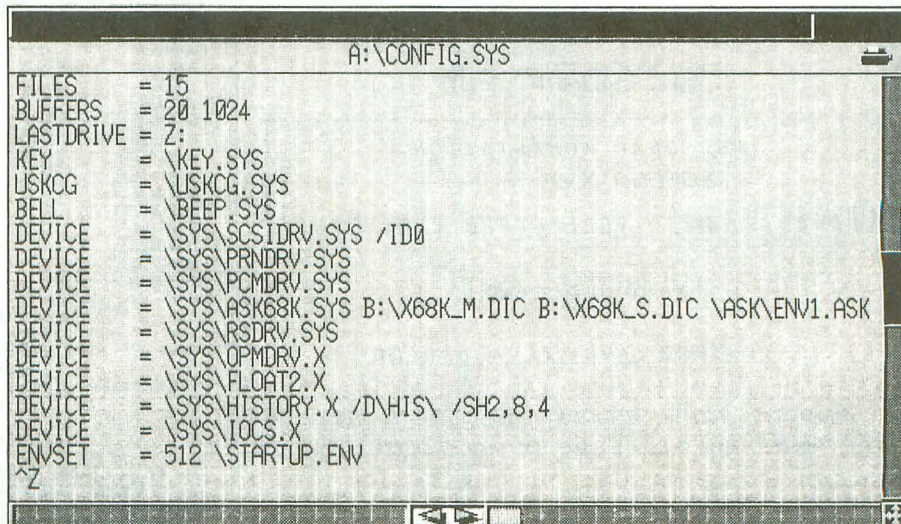
●文書ファイル

一方の文書ファイルだが，こちらはアプリケーションに依存するファイルだ。ちょっとワープロの文書を思い浮かべてみてほしい。ワープロの文書には，単なる文字の羅列だけではなく，1行の文字数や文字間隔，行間隔，1ページあたりの行数など書式に関する情報が必要だ。これらは必ずしも文書ファイルの中に保存しておかなくても，ワープロ本体の側で指定することはできる。とはいっても，データを読み込むたびに書式を設定しなおすのは結構うっとうしいから，文書ごとに保持しておきたいものだ。

どうしても文書ファイル内に持っておかなければならない情報としては，罫線があったり，文字にも強調や斜体などの装飾があったりといったものがある。そして，当然これらの情報はワープロソフトごとに異なっている。ワープロの文書以外のデータベースソフトやスプレッドシートなどのデータファイルも，文字情報を扱っていながらアプリケーション独自の機能によって管理されたものと考えればよいだろう。

テキストの内容を見る

基本となるテキストファイルについても

```
A:\CONFIG.SYS
FILES     = 15
BUFFERS   = 20 1024
LASTDRIVE = Z:
KEY       = \KEY.SYS
USKCG     = \USKCG.SYS
BELL      = \BEEP.SYS
DEVICE    = \SYS\SCSIDRV.SYS /ID0
DEVICE    = \SYS\PRNDRV.SYS
DEVICE    = \SYS\PCMDRV.SYS
DEVICE    = \SYS\ASK68K.SYS B:\X68K_M.DIC B:\X68K_S.DIC \ASK\ENV1.ASK
DEVICE    = \SYS\RSDRV.SYS
DEVICE    = \SYS\OPMDRV.X
DEVICE    = \SYS\FLOAT2.X
DEVICE    = \SYS\HISTORY.X /D\HIS\ /SH2,8,4
DEVICE    = \SYS\IOCS.X
ENVSET    = 512 \STARTUP.ENV
^Z
```
図1　タイプ.XでCONFIG.SYSを表示

う少し具体的に見ていこう。とりあえず誰もが持っているシステムディスクを覗いてみる。ルートディレクトリにあるテキストファイルは，CONFIG.SYSとAUTOEXEC.BATの2つ。そこで，CONFIG.SYSをタイプしてみよう。以前はコマンドモードから，

A>type config.sys

として，画面に表示するのが通例であったが，SX-WINDOWのおかげでファイルアイコンを選択してポップアップメニューから内容表示（タイプ.X）を実行すれば図1のように表示することができる。基本的には同じものだが，タイプ.Xでは最後に^Zというファイルの最後を表すコントロールコードが表示されている。通常は見えないようになっているものだからあまり気にすることはないが，ツールによって同じテキスト情報でも表示の扱いが違っていたりすることは心に留めておいたほうがよいだろう。

同様に，AUTOEXEC.BATやBINディレクトリに入っているED.HLP（ED.Xのヘルプファイル），ASKディレクトリのENV1～5.ASK（仮名漢字変換のキー割り付けを指定する環境ファイル）などを覗いてみよう。いずれもテキストファイルだからタイプできるはずだ。Human68kのマニュアルを参考にし，だいたいの内容がわかったら，ちょくちょく書き換えて自分なりのシステム環境を考えてみるとよいだろう。

特に，CONFIG.SYSは自分で環境を整備しようという人には避けて通れないファイルだから，必要に応じて書き換えられるようにしたい。ファイルを書き換えるにはなんらかのエディタを使うことを勧める。ワープロでもかまわないのだが，この手のものを編集するときは自動的にバックアップファイルを残してくれるエディタのほうがなにかと安心だ。失敗したと思ったら，さっさとバックアップファイル（CONFIG.BAKなどとなってる）をリネームして復活させればいい。

テキストファイルの種類

さて，システムディスクに入っているテキストファイルは割と特別な役割を持つものが中心だ。もうちょっと一般的なテキストファイルにはどんなものがあるだろう。多くの場合，テキストファイルの種類によって決まった拡張子をつける習慣がある。

～.DOC　ドキュメントファイル。なんらかの原稿が書かれていると思ってよい。ディスクに入っているプログラムの解説などにも多い。特にREADME.DOCとかあれば，迷わず読んでみることだ。

～.S　アセンブラのソースプログラム。これが読み書きできるようになれば怖いものはない。村田氏のX68000マシン語プログラミング入門で勉強しよう。

～.C　Cのソースプログラム。同じくC言語の知識を必要とする。なお，配布されたプログラムにバグがあっても，ソースファイルとCコンパイラがあれば，ソースの修正点を書き換えてコンパイラしなおせばよい。

～.BAS　BASICのプログラムテキストだ。OS上で動くBASICが標準で付いているのはX68000とAMIGAぐらいのもの。

～.HLP　主にアプリケーションが呼び出すヘルプファイルがこれ。

とりあえずこのあたりだが，PC-9801などのDOSユーザーと交流があれば，次のものも押さえておきたい。

～.TXT　いかにもテキストファイルであると自己主張している。アプリケーションが自分のデータファイルをテキストファイルに変換して出力する際によくこの拡張子がつけられる。

～.JXW　天下の（？）一太郎さんの出力するテキストファイル。

といったところだ。

＊

テキストファイルをいくつか覗いてみたところで，こんどはワープロなどの文書とどのように違うかを見てみよう。

WP.Xと～.SWPファイル

X68000には初代機以来，標準で日本語ワードプロセッサWP.Xが付属している。今回の新製品X68000XVIではある程度のバージョンアップがなされたが，ワープロ自体の話はまた別の機会にということで，ここではファイル関係の話をまとめておこう。

WP.Xの文書ファイルは～.SWPという拡張子が使われる。サンプルとして，ワープロディスクのQUICKSTARTに入っている「自己紹介.SWP」という文書ファイル

図2　WP.Xの印刷例

見はてぬ夢の象徴

PERSONAL WORKSTATION
X68000 XVI series
Power to make your dream come true

X68000 新たなる展開

クロック周波数 16MHz MPU68000を搭載

X68000は今、遥な世界へ翔びたつために、新たなる翼を手にいれました。
クロック周波数16MHzのMPU68000は、快適な環境を現実のものにします。

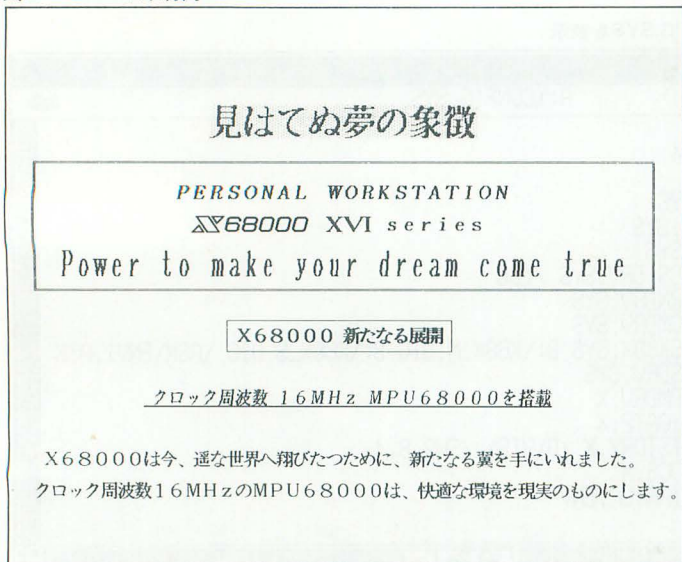



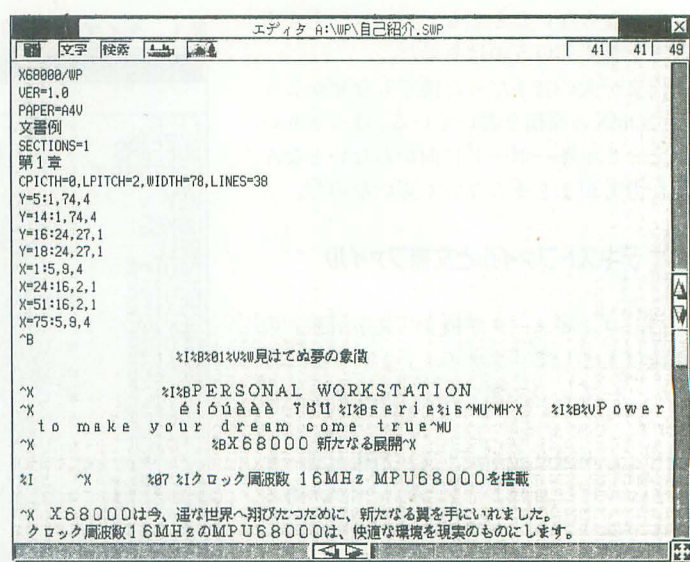
図3　ワープロ文書のしくみ

を読み込んでみよう。

図2はX68000XVIに付属してくる「自己紹介.SWP」のプリンタ出力だ。内容はX68000の紹介文だが，WP.X自体の紹介も兼ねているため，強調，斜体，縦倍角，4倍角，装飾，罫線，外字のオンパレードである。ただの文字が並んでいるだけのテキストでないことはひと目でわかるだろう。しかもこれはWP.X上のひとつの章の内容なのだ。

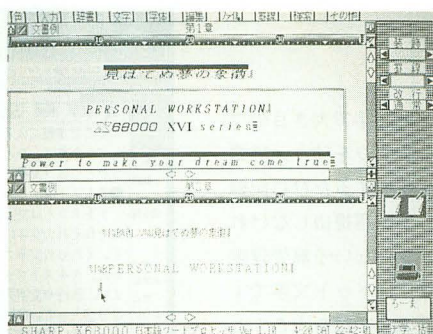
一般にこういったワープロの文書ファイルのようなものは，そのワープロ上でなければ扱えない専用のデータ形式になっていて，OS上のTYPEコマンドや一般のテキストエディタでは読み込めないことが多い。Hyperwordなどもそうだ。その場合，OS上からちょっとファイルの中を覗いてみたいとか，修正したいといったことができない。ちなみに，一太郎の文書ファイルは，通常のテキストファイルと罫線や装飾などの付加情報（アトリビュートなどという）が別のファイルに分離されており，～.JXWというテキストファイルだけをDOS上で自由に扱うことができる。それでも，アトリビュートの部分はいじれない。

もちろん，WP.Xでもファイル出力という機能を利用すれば，文書から罫線や装飾などの付加情報を取り除いたごく普通のテキストファイルに変換して出力することができる。また，同様に章をひとつ開いて，そこにテキストファイルを読み込むファイル入力という機能もある。

ところが，WP.Xにはそれよりもデータ形式自体にちょっと面白い特徴がある。実はWP.Xの文書ファイルはふつうのテキストファイルと同様に編集可能である。文字の種類や装飾情報はテキスト情報で扱える特殊文字（コントロールコード）に割り当てられているのだ。どういうことか，説明するよりも図を見てもらったほうが早い。

図3は，新しいSX-WINDOWのエディタ.XでWP.Xの文書ファイルである「自己紹介.SWP」を読み込んだところだ。情報量を多くするため12ドット文字を使い，行間隔を3ドットに設定している。

まず，1行目のX68000/WPからX=～までがヘッダ部分。用紙サイズ，文書名，章番号と章名，印字のピッチ（文字間隔，行間隔），1行の文字数，1ページの行数，そして続く座標は罫線情報だ。

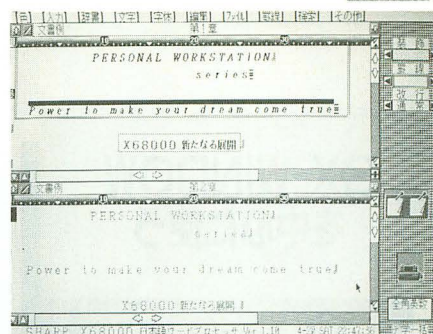
そしていよいよ各章の内容が始まる。冒頭の「見はてぬ夢の象徴」には多くの特殊文字がついているが，いずれも文字に対する付加情報だ。順番にいくと，%Iは斜体，%Bは強調，%01は装飾番号，%Vは縦倍角，%Wは横倍角，といったぐあいだ。

また，WP.Xでは印刷時の改行幅も行ごとに指定できる。図3の例では，^MUは上合わせ改行，^MHは1/2改行となっている。

とまあ，このような仕組みがわかっていれば，ワープロの文書をエディタなどで読み込んで得体の知れない特殊文字がたくさん出てきてもうろたえる心配はない。ここで，文書内容に変更を加えることも容易だろう。SX-WINDOWの環境がよくなってくると，ワープロの文書をちょこっと参照したりするためにウィンドウから抜け出すのはたいへん億劫になる。WP.Xだってそこそこ大きなシステムだから起動にも多少の時間がかかるからね。もちろん，コマンドモードになれている人も同様だろう。

それから，WP.Xを愛用している人もこれがわかっていると便利なはずだ。というのも，ワープロよりもエディタのほうが都合のいい機能があることも多いからである。文字列の置換などがいい例だ。

たとえば，ながーい原稿で，「、」を「，」に変更したくなったとしよう。WP.Xの置換機能では検索文字列がなくなるまで無条件に置換するということができないから，エディタを使いたくなる。通常はワープロの文書をエディタに持っていくにはファイル出力を考えるだろうが，それではせっかくワープロ上で指定したアトリビュート情報が消えてしまう。が，ファイル出力などしなくても文書のままエディタに持ち込み，置換作業を達成したのち，平然とワープロに戻って文書呼び出しを行えばことはすむわけだ。

置換以外にもエディタのほうが便利なことの代表としてはキーマクロがある。キーマクロとは一定のキー操作を記憶させ，必要な回数だけ繰り返させる機能である。

表形式の文書を編集していて，項目の位置を修正したいと思ったら，カーソル移動とスペース挿入などの繰り返しを延々とやる必要があるが，そういうときにはエディタのキーマクロがらくちんでよい。ちなみに，SX-WINDOWのエディタ.Xなら，ひとつのウィンドウでキーマクロを実行させながら，別のウィンドウで編集作業ができるというメリットもある。時間のかかるキーマクロの際には，疑似マルチタスクも結構おいしい。

ファイル出力で外字が消える

外字を削除した場合

ファイル出力と外字

先ほども触れたように，WP.Xはファイル出力またはファイル入力によって通常のテキストファイルを扱えるようにしている。これにより，他機種（MS-DOSマシン）とのファイル互換も結構うまくいく。一太郎の～.JXWファイルもファイル入力で問題なく読み込めるし，逆にWP.Xの文書もファイル出力で～.JXWとつけておけば一太郎で利用できる。

ただし，残念ながらうまくいかない面もある。外字が入っている場合だ。WP.X上で作成した外字が文書中にあると，ファイル出力の際にその手前までしか正しく出力してくれない。試しに，例の「自己紹介.SWP」の文書をファイル出力し，それを再びファイル入力で第2章に読み込ませたのが写真1だ。外字で作られた「X68000～」の手前で途切れてしまっている。写真2のように元の外字の部分を削除して同様の手順を踏むと，以降の部分も読めるようだ。

＊

というわけで，付属の日本語ワードプロセッサWP.XとSX-WINDOWなどの環境をベースにして，文書ファイルやテキストファイルを扱う際に，念頭に置いておきたいことがらをまとめてみた。初心者がパソコンに接する際には，まずこうしたテキストファイルの扱いに慣れることが，OS上の操作を理解し，使いやすいシステム環境を構築するのに役立つのではないだろうか。

SX-WINDOWを中心に使う

僕はX68000を会社に持ち込んで自分の机に置いている。もちろん，時間外にX68000ならではのゲームソフトを周囲にひけらかすのも一興だが，ちゃんと仕事に使っているのだ。以前はワープロがほとんどだったが，現在はSX-WINDOWをベースにしている。

ハードウェアは初代X68000だが，ソフトは原稿を書く条件で最新のシステムを使わせてもらっている。したがってSX-WINDOWはVer.1.10だ。メインメモリが4Mバイトにハードディスク(40Mバイトのやつね)。プリンタは持っていないが，なぜか24ドットの熱転写プリンタが転がっていたのでちょっと拝借している。

で，いままでと劇的に使用環境が変わったのは，なんといってもエディタ.XとSX-WINDOW全体の高速化によるところが大きい。

仕事場では企画書や管理文書などの書類を作成したりするのだが，急ぎで書類をまとめなきゃならない日でも，机に座っていれば随時細かい割り込み仕事が入ってくる。連絡事項をメモしたり，ファイルをコピーしたり，別の書類の提出期限が近づいてきたりする(たいていは忘れいたふりをするんだけどね)。なによりも，メモしておきたいような思いつきというのは時をわきまえない。別のメモを参照したり更新したり，ものごとは同時進行的に処理していかなければならないのが世の常だ。

そこで，SX-WINDOWの登場だ。さっきいったエディタ.Xは同時に何枚ものウィンドウをそれぞれの内容に応じた書式で開いておける。現在メインで作業中のウィンドウは16ドット文字で大き目のウィンドウを使う。経費が発生するたびに記録して毎週提出しなければならない予算管理表は，12ドット文字で1行の文字数を大きくとる。連絡事項や電話番号が記録されたウィンドウも当然あるし，その他何種類ものメモウィンドウが画面に重なっている。

あまりにも便利なAGAIN

重なったエディタウィンドウはHOME/CLRキーでほかのウィンドウとは関係なしにページをめくれ，素早くほかのウィンドウを参照したり，ウィンドウ間のカットアンドペーストができる。あるウィンドウでキーワードをマウスで拾って別のウィンドウで検索をかけるといったことも簡単。もちろん，SX-WINDOW上だから同時にさまざまなファイル処理も可能だ。この便利さは筆舌につくしがたいものがある。それから，システムアイコンで終了時の画面を保存しておくようにするとよい。途中でSX-WINDOWを抜け出してもエディタ上の書式などを再度設定する必要はない。

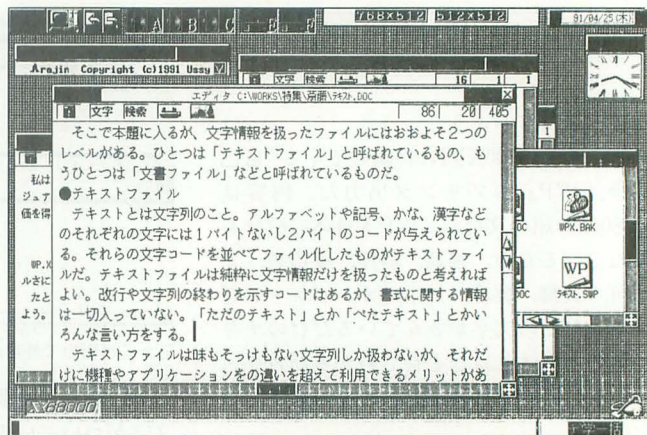
息がとぎれたら，5月号付録ディスクの「信州」を開く。マルチウィンドウだから途中で終了する必要はない。アイコン化を選択すれば，タイトルバーだけにしておくこともできる。ボスが来たモードなんてそもそも不要だ。また，画面を512×512ドットモードにして「SX風船」を10個ほど飛ばしてみると気分が落ち着く。マウスでパンパン割るのがまた楽しい。

新しいプリンタドライバもいい。電話番号などのメモは16ドット文字でイメージ印字すれば細かい文字で印刷され，システム手帳のリフィルにも都合がよい。

ところで，日頃ワープロを使っている人が普通にASK68K(標準の日本語変換システム)を使って戸惑うのは，変換キーを押したら必ず確定キーを押さないと次の文字が入力できないことだろう。これを次の文字を入力したら自動的に確定して進めるようにするには，ASK68Kが参照するの環境ファイルを書き換える必要がある。

システムのASKディレクトリにあるENV1.ASKを選択してエディタ.Xを起動する。お目当ての箇所は最後の行で，

DEFCONT＝0 → DEFCONT＝1

と書き換えておこう。ファイルネームを変え

て保存する場合は，CONNFIG.SYSの参照している環境ファイル名を書き換えるのを忘れてはいけない。

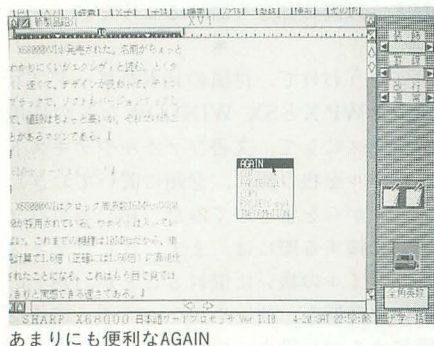
●WP.Xだって捨てたものじゃない

さてと，このままだとWP.Xの立つ瀬がないようだが，決してそんなことはない。表などはやっぱり罫線などが欲しい。そこで，WP.Xほうであらかじめ罫線を引いた表のフォーマットを作っておき，あとからテキストを流し込んで印刷するのがベスト。WP.Xの罫線は文字画面と独立しているからこういう技も効くのだ。

一部ではX68000のワープロは仕事には使えないなんていう困った輩がいるようだが，WP.Xは指向性の面ではかなり志の高いワープロだ。WP.Xが優れている点のひとつはユーザーインタフェイスのシンプルさにある。まず，マウスで範囲を指定してカットアンドペースや再変換(これはエディタじゃできない)ができる。1行の文字数も章ごとに決められる，それもマウスでゲージをびよーんと動かすだけ。普通のワープロなら印刷機能を選んで書式設定を選んでと大変な苦労をするところだ。

そして極めつけがポップアップメニューの「AGAIN」だ。たとえば，見出しを強調文字にしたいとき，最初はマウスで指定して，プルダウンメニューの［文字］から「強調」を選ぶわけだが，次の見出しを指定したら今度は右ボタンを1回押すだけでよい。ポップアップメニューは右ボタンが押された場所で開き，マウスカーソルはちょうど「AGAIN」の上にいるからだ。

＊

今回は手放しの喜びようだが，多くの皆さんはまだまだSX-WINDOWのVer.1.10を持っていないはずだ。シャープからは多少のチェック期間をおいてパッケージされるはずだからぜひともメモリとハードディスクを用意して出迎えることをお進めしたい。

吾輩はX68000である
【第2回】

いでよ！ 文字たち

Izumi Daisuke　泉　大介

御仁は文字を出すのに熱中する
とはいえ，文字はいずこから現れるのか
吾輩が秘密を解き明かそう

前回は吾輩がなぜテレビ受像機(モニタディスプレイ)に文字を表示できるのかを，テキストVRAMの構造をからめながらお話しした。前回御仁は吾輩が持っている「A」の文字を自分で直接テキストVRAMにセットして表示することに熱中していたが，吾輩が実際に文字を表示するときにはどうやっているのか。吾輩が保持している「A」の文字はいったいどこにあるのか。今回はこの辺りをお話しすることにしよう。

文字データの所在

吾輩が8，12，16，24ドットの4種類の文字セットを持っていることは前回お話しした。吾輩はこれらの文字データを，ドットの点・消燈を意味する1と0を羅列した形で蓄えている。どこに？　もちろんメモリの中にである。ご存じのようにメモリにはROMとRAMの2種類がある。ROMは電源を切っても情報が消えないメモリ，RAMは通電している間だけ情報を保持できるメモリである。文字データがROMに収めてあることはいうまでもない(さもないと，電源を切った瞬間に文字データがすべて消えてしまう)。

図1をご覧いただきたい。これは吾輩のメモリが，どのように使われているかを表す「メモリマップ」と呼ばれる図である。いわば吾輩の身体解剖図のようなものであり，いささか恥ずかしいので，あまりジロジロと眺めないでいただきたい。

公開ついでに補足しておくと，まず最初，アドレス000000_Hからは，吾輩の動作の基幹をなすHuman68kが使用している。Human68kはディスクドライブをA，Bなどの名前で扱えるようにしたり，アプリケーションプログラムをディスクから起動できるようにするプログラムで，読者諸兄が馴染み深いビジュアルシェルやCOMMAND.Xも，このHuman68kがなければ動かない。

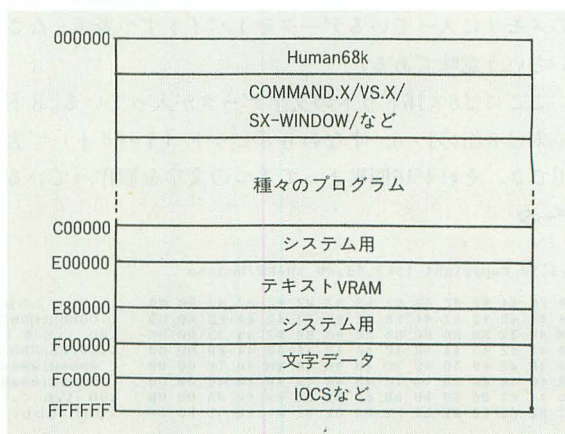
続くメモリにはCOMMAND.Xなどが収められる。諸兄が起動するアプリケーションプログラムは，さらにこのあとに続いて収められることになる。ここはメインメモリと呼ばれており，メモリの増設を行うとこのエリアが拡がって，より大きなプログラムを実行したり，アプリケーションの中でより多くのデータを扱うことができるようになる。$C00000_H$までの間が点線になっているのは，諸兄がメモリをどれだけ増設しているかでメインメモリの量が変わるからである。

C00000～$DFFFFF_H$はシステム用と表記してある。ここにはグラフィックを表示するためのグラフィックVRAMなどが収められているが，今回の話題には関係がないので省略した。E80000～$EFFFFF_H$も同様である。

$F00000_H$以降が今回の話題の主役であるROMが収められているエリアとなっている。F00000～$FBFFFF_H$が文字データが収められている領域で，ここは特にCGROM(シージーロム)と呼ばれている。CGというのはコンピュータグラフィックではなく，キャラクタジェネレータ(Character Generator：文字生成部)を略したものである。

続く$FC0000_H$以降にはIOCS(アイオーシーエス)と呼ばれるサービスルーチンが収めてある。これはInput Output Control Systemを略したもので，入出力をコントロールするシステムという意味になるだろうか。吾輩のようなコンピュータにとって，入力とはキーボードやマウスなど本体の外からデータを受け取ることであり，

図1　吾輩のメモリマップ

アドレス	内容
000000	Human68k
	COMMAND.X/VS.X/SX-WINDOW/など
	種々のプログラム
C00000	システム用
E00000	テキストVRAM
E80000	システム用
F00000	文字データ
FC0000	IOCSなど
FFFFFF	

出力とはディスプレイやプリンタなど本体の外にデータを送り出すことを意味している。

本来，入出力は吾輩のハードウェアを直接制御するプログラムを作らなければ行うことができず，ソフトウェアの知識だけでなくハードウェアの知識も要求される非常に面倒な部分である。IOCSはこういった面倒なプログラムをあらかじめ用意したもので，これさえあれば，ハードウェアの知識はなくとも吾輩を自由に操ることができる。

以上で吾輩のメモリマップの説明は終わりである。文字データだけでなく，Human68kやCOMMAND.X，IOCSといったプログラムまでがメモリに入っていることに驚かれたかもしれない。吾輩はメモリに入っているデータを扱う。そしてメモリに入っているプログラムを実行する。これこそが，かのフォン・ノイマン大先生が提唱した，「プログラムもデータもメモリに入れちゃえ」方式である。

ちなみに，FFFFFF$_H$となっているアドレスの最終番地が吾輩が扱えるアドレスの限界となる。つまり，吾輩は1000000$_H$バイト(FFFFFF$_H$+1)のデータ（やプログラム）を扱えるわけだ。人間は大きな単位を扱う場合に，KだのMだのといった記号を使うが，1K（イチケー）バイトは1024バイト，1M(イチメガ)バイトは1024Kバイトとされている。吾輩の扱えるデータ量1000000$_H$を1M(100000$_H$)で割れば答えは10$_H$。吾輩が16Mバイト(10$_H$Mバイト）のメモリを扱うことができると言われるのはこのためである。

CGROMを覗く

では吾輩が，実際にどのような形で文字データを保持しているのかを見ていこう。図2はROM内の「A」の文字を入れてある部分をデバッガで表示したものである。XCなどに付属のデバッガを持っていない人は，1991年1月号の付録ディスクにSX-WINDOW開発ツールとして入っているものを使ってもよい。「-」に続いて入力されている「ds f3ac10」というコマンドは，F3AC10$_H$以降のメモリに入っているデータを1バイトずつ表示しなさいという意味である。

ここには8×16ドットの文字データが入っている。8ドットは8桁の1・0，すなわち8ビット（1バイト）で表現でき，それが16個集まって1つの文字を形作っていることは前回お話しした。つまり，頭に00F3AC10と表示されている行に表示されている16個の16進数が「A」のデータである。確認してみよう。まずデータを2進数に変換し，1を点燈，0を消燈の合図とすれば「A」の文字が表れる。図3を参照されたい。前回やったように，このデータをテキストVRAMに順にコピーしていけば画面に「A」を表示できるという寸法である。続くF3AC20$_H$には「B」のデータ，F3AC30$_H$には「C」のデータが入っている。図3の要領で確認してみていただきたい。

●御仁悩む

CGROMはF00000～FC0000$_H$と，実に768Kバイトもの大容量である。目的の文字のデータをただ当てずっぽうで探し出すというわけには，とてもじゃないがいくものではない。御仁も自分の手でひとしきりテキストVRAMにデータをセットして遊んだあと，CGROMを覗いてみようと思い立ったらしいが，この事実に気づいてハタと手を止めてしまった。デバッガの「d」コマンドでCGROMを眺めてみたまではよかったのだが，図2のようなデータ並びが表示されたのでは，いったいなんの文字なのかわかろうはずもあるまい。

それなら，というわけで次に御仁が行ったのは，「A」の文字データの最初の部分である

　00 10 28

というデータ並びをF00000～FBFFFF$_H$（FC0000$_H$−1）の中から見つけ出すことである。これにはデバッガの「ms」コマンドが有効だ。「ms」というのはMemory Search（メモリ検索）の略である(たぶん)。ここでは1バイトのデータ並びを検索するため，「ms」コマンドの後ろに「s」の文字を付けて「mss」とすればいい。

結果は図4のとおりである。「00 10 28」というデータ

図3　CGROMから文字を作る

```
00 → 00000000 → ○○○○○○○○
10 → 00010000 → ○○○●○○○○
28 → 00101000 → ○○●○●○○○
44 → 01000100 → ○●○○○●○○
82 → 10000010 → ●○○○○○●○
82 → 10000010 → ●○○○○○●○
82 → 10000010 → ●○○○○○●○
82 → 10000010 → ●○○○○○●○
FE → 11111110 → ●●●●●●●○
82 → 10000010 → ●○○○○○●○
82 → 10000010 → ●○○○○○●○
82 → 10000010 → ●○○○○○●○
82 → 10000010 → ●○○○○○●○
82 → 10000010 → ●○○○○○●○
00 → 00000000 → ○○○○○○○○
00 → 00000000 → ○○○○○○○○
```

図2「A」の文字データ

```
X68k Debugger v2.10 Copyright 1987,88,89 SHARP/Hudson
-ds f3ac10
00F3AC10  00 10 28 44 82 82 82 82 FE 82 82 82 82 82 00 00   ..(Dbb.bb※.
00F3AC20  00 F8 44 42 42 42 44 78 44 42 42 42 44 F8 00 00   ..DBBBDxDBBBD...
00F3AC30  00 38 44 82 80 80 80 80 80 80 80 82 44 38 00 00   .8D \\\ 8..
00F3AC40  00 F8 44 42 42 42 42 42 42 42 42 42 44 F8 00 00   ..DBBBBBBBBBD...
00F3AC50  00 7E 40 40 40 40 40 7C 40 40 40 40 40 7E 00 00   .~@@@@@|@@@@@~..
00F3AC60  00 7E 40 40 40 40 40 7C 40 40 40 40 40 40 00 00   .~@@@@@|@@@@@@..
00F3AC70  00 38 44 82 80 80 80 80 8E 82 82 82 46 3A 00 00   .8D \:b :..
00F3AC80  00 82 82 82 82 82 82 FE 82 82 82 82 82 82 00 00   .bbb.bbb..
```

図4　文字データのアドレスを探す

```
-mss f00000 fbffff 00 10 28
00F3A2EF  00F3A768  00F3AC10  00F3ADE0
-
```

並びは，F00000～FBFFFF$_H$の中に4カ所もある。ちなみに「00 10」だけだと膨大な数にのぼり，とても探せたものではない。そもそも，「A」という文字のデータが入っているアドレスを探すのは，その文字データが欲しいからである。にもかかわらず，欲しいはずの文字データからアドレスを探そうというのは本末転倒もいいところ。これはいただけない。

そのまま諦めるかと思われたが，すんでのところでショックから立ち直ったらしい。キーボードの上で凍りついていた手は，すぐさまプログラマーズマニュアルに伸び，ページを繰り始める。吾輩のIOCSを使うことを思い付いたのである。

IOCSに収められている数々のサービスルーチンのなかには，文字データの格納アドレスを調べるものが用意されている。ただ，IOCSを利用するには，ちょっとばかり吾輩の頭脳であるMC68000のことを知っていなければならない。囲みにしておいたので参照されたい。では御仁のお手並み拝見といこう。

●御仁のプログラム

図5をご覧いただきたい。これが御仁の行った操作である。御仁の操作は，まず「P」コマンドでメモリの使用状況を調べることから始まった。最初に表示されている$000796C0はデバッガが入っているアドレスを意味し，次の$000A70A0がユーザーがプログラムを作っていいアドレスを示している。つまり$000A70A0以降は空いているので，自由に使っていいという合図である。ここに表示される数値は諸兄のマシンの使用状況に応じて変わるので，違うからといってがっかりしないでいただきたい。要は空いているところにプログラムを作ればいいだけである。いくら自由に使っていいといわれても，メインメモリを越えることはできない。1Mしかメモリを搭載していないマシンでは1M－1＝100000$_H$－1＝FFFFF$_H$まで，2M搭載しているマシンでは1FFFFFまでである。

御仁は空きだと示されたA70A0$_H$番地から，IOCSを利用するプログラムを作成することに決めたようだ。

-a a70a0

と入力しているのは，A70A0$_H$からプログラムを作成するというデバッガへの指示である。続いて，

000A70A0　ori.b　#$00,D0

とあるのは，現在A70A0$_H$には，「ori.b #$00,D0」という命令が入っていることをデバッガが表示しているのである。気にせずプログラムを書き込んでいるのがその次の行の，

move.w　#$41,d1

である。D1に41$_H$をセットする命令を，ここに書き込んだわけだ。

41$_H$というのは「A」という文字に与えられたASCII（アスキー）コードと呼ばれる番号である。デバッガでは文字を「'」（シングルクォート）でくくれば自動的にそのASCIIコードに変換されるので，この行は，

move.w　#'A',d1

でも同じことだ。どちらかというとこのほうがなにを行っているのかわかりやすくていいかもしれない。続く行でD2，D0にもデータをセットしている。

IOCSを利用するときは，このようにレジスタに必要なデータをセットする必要がある。御仁が利用しようとしている「文字の格納アドレスを調べる」サービスでは，

D1.w)　調べたい文字のコード

D2.l)　文字のサイズ

6：12×12，6×12ドット

8：16×16，8×16ドット

12：24×24，12×24ドット

D0.l)　16$_H$

をセットしなければならない。これはプログラマーズマニュアルに書いてあるとおりである。レジスタ名の後ろに「.w」とか「.l」がついているが，これはデータをワードでセットするか，ロングワードでセットするかを指示しているものだ。これは仕様なので，指定されたとおりに「move.w」「move.l」を使い分ければいい。

御仁のプログラムと照らし合わせてみよう。D1にセットされているのは「A」のASCIIコード，D2にセットされているのは8なので，図5は「Aの8×16ドットフォントが格納されているアドレスを知る」ということになる。

プログラムの最後は，

trap　#15

となっている。これはIOCSを使うためのおまじないのようなもので，レジスタに必要なデータをセットしてこの命令を使えば，IOCSのサービスが利用できるようになっているのである。IOCS作法として覚えておくといいだろう。

aコマンドで始めたプログラム作成は，「.」を入力してリターンキーを押せば終了する。

●文字データの格納アドレスを知る

作成したプログラムを実行すれば，指定した文字の格納アドレスがD0.lに入れられることになっている。プログラムの実行は「g」コマンドだ。

g＝プログラムのアドレス

と入力すればプログラムが実行される。が，その前にや

図5　文字データのアドレスを調べるプログラム

```
-p                                          ← メモリの空きを調べる
debug program from $000796C0
user  program from $000A70A0                ← A70A0H番地以降が使える
-a a70a0                                    ← A70A0H番地からプログラムを作成する
  000A70A0       ori.b    #$00,D0
                 move.w   #$41,d1           ← D1に、調べたい文字コードをセット
  000A70A4       ori.b    #$00,D0
                 move.l   #8,d2             ← D2に文字サイズをセット
  000A70AA       ori.b    #$00,D0
                 move.l   #$16,d0           ← D0に、サービス番号16Hをセット
  000A70B0       ori.b    #$00,D0
                 trap     #15               ← IOCSを使う際のおまじない
  000A70B2       ori.b    #$00,D0
                 .                          ← プログラム作成の終了
```

っておかなければならないことがある。それは，プログラムの実行を中断するアドレスの指定である。これがないと，メモリ内のデータをプログラムだと見なして延々と実行し続けるという間抜けな事態が発生してしまう。

なぜそんなことが起きるのか。理由は単純。吾輩にとっては，プログラムもデータも同じものだからなのである。たとえば図4のプログラムの先頭で，A70A0$_H$番地には，

ori.b　　#$00,D0

というプログラムが入っていると表示されている部分がある。このときA70A0～A70A3番地には，00000000$_H$というデータが入っているだけである。もしかするとこれは0というロングワードのデータなのかもしれない。しかし，上のようなプログラムだと見なすことも可能なのである。いずれにせよ，吾輩にとっては「メモリに入ったデータ」以上の意味はもたない。それをプログラムだと見なして実行させるか，データとして扱わせるかは諸兄自身である。いったん実行しなさいといわれれば，吾輩はメモリにセットされたデータをプログラムだと見なしてどこまでも実行していく。悲しいかな，それがコンピュータとしてもって生まれた性なのである。

したがってデバッガでプログラムを作成した際には，「実行はここまで」というアドレスを指示してもらわなければならぬ。図6をご覧いただきたい。これは図5の続きである。「b」コマンドは，プログラムの実行を中断するポイントを設定するもので，b0～b9の10個の中断ポイント（ブレイクポイント）を設定しておける。

設定方法は簡単で，設定したいポイント名に続けてアドレスを入力するだけである。ここではA70B2$_H$番地を，b0というポイント名で設定している。なお，単に「b」とだけ入力すれば，現在設定されているブレイクポイントの一覧を見ることができる。プログラムは，ブレイクポイントを設定したアドレスに入っている命令を実行する直前で停止するので，プログラム入力の最後に「.」を入力した際に表示されていたアドレスを設定すればいい。

ブレイクポイントを設定したら，前述のgコマンドでプログラムの実行である。A70B2$_H$で実行は中断され，そのときのレジスタの内容が表示される。行頭に「D」と表示された行にD0～D7の内容が，行頭に「A」と表示された行にA0～A7の内容が表示されている。文字データの格納アドレスはD0.lに収められている。D0.lはF3AC10$_H$。図2で内容を確認したあのアドレスである。また

D1.wには，

　文字データの横方向のバイト数－1

が，そしてD2.wには，

　文字データの縦方向のドット数－1

がセットされている。「A」は横8ドット（=8ビット=1バイト）なのでD1.wは1－1で0。縦は16ドットなのでD2.wは16－1=15=F$_H$となっている。

CGROMを利用した文字表示

文字データが格納されているアドレスがわかれば，それを表示してみたいと思うのは自然の理であろう。御仁が次に取り掛かったのも表示部分である。文字データが格納されているアドレスからデータを1つひとつ取り出し，それを先月の要領でテキストVRAMにセットしていけば文字は表示できるが（実際吾輩もそうやって文字を表示しているのだが），さすがに御仁も今回はIOCSを使うようである。

IOCSのNo.18$_H$は，No.16$_H$と対になったサービスである。その使い方は，

D1.w)　データの横方向のバイト数－1
D2.w)　データの縦方向のドット数－1
D3.l)　後述
A1.l)　データの先頭アドレス
A2.l)　表示するテキストVRAMのアドレス
D0.l)　18$_H$

となっている。D1，D2はNo.16$_H$のIOCSサービスでセットされるデータをそのまま利用し，A1はD0をコピーして使えばOKである。D3はいささか複雑なデータで，

　128－（D1.w＋1）

となっている。128というのは画面の横幅のバイト数である。吾輩のテキストVRAMは横1024ドットある。1アドレスに横8ドット分のデータ（=1バイト）をセットできるため，横1024ドットなら1024÷8=128となる。ここから表示する文字の横方向のバイト数を引いたものがD3.lにセットする値だ。

では御仁の作ったプログラムを見ていただこう。図7である。図7は文字を表示する全プログラムとなっているが，文字データの格納アドレスを得るところまでは図5と同じである。

続いてD3.lに127をセットしているが，これは上の式のカッコを外し，128－1を先に計算したためである。そ

図6　プログラムの実行

```
-b0  a70b2                                    ← A70B2Hでプログラムの実行を中断
-g=a70a0                                      ← A70A0Hに入れたプログラムを実行

     break at 000A70B2                        ← A70B2Hで実行が中断された
PC=000A70B2 USP=000857E4 SSP=000067F2 SR=0000 X:0  N:0  Z:0  V:0  C:0
D  00F3AC10 00080000 0000000F 00000000  00000000 00000000 00000000 000
A  00000000 00000000 00000000 00000000  00000000 00000000 00000000 000
ori.b   #$00,D0
-q                                            ← デバッガを終了
```

のあとsub命令を使ってD3からD1を引いている。実際にはD3.l－D1.wを計算しなければならないところだが，そのような命令はない。どうせ24×24ドットフォントを取り出した場合でもD0.wは2にしかならない（24÷8－1＝2）ので，計算は127－0～127－2のいずれか，つまり，

```
    0000007F          0000007F
 －)    0000    ～  －)    0002
```

のいずれかを計算するだけである。上4桁は計算に関係ない。御仁はここではsub.wで計算することにしたようだ。引き算の結果はD3.wに入る。

A1.lにD0.lをコピーしているところでは，move命令のバリエーションのひとつであるmovea命令を使っている。A0～A7レジスタにデータをセットする場合には，movea命令を使わなければならない。また，

```
movea.l  #d0,a1
```

となっていないことに注意されたい。「#」は数値を意味するマークだった。D0は数値ではなくレジスタ名なので「#」は不要なのである。moveとmoveaの違いは，御仁がコラムを書いてくれたので参照されたい。

A2.lには文字の最初の8ドットを入れるアドレスをセットする。ここでは先月に引き続き，マウスプレーンに文字を表示することにしたようだ。最後にD0.lにサービスの番号をセットしてtrap命令である。

実行してみて，御仁はいたく満足気である。データを1つひとつ手でセットしていた前回と異なり，今回はちょっとしたプログラムだけで表示できるのだから当然である。しかも，A70A0$_H$を，

```
move.w #'泉',d1
```

に変更するだけで「泉」の文字を表示でき，さらにA70A4$_H$を，

```
move.l #12,d2
```

とすれば24ドットフォントで表示できる。こんなに楽なことはない。御仁は次々と文字を表示しては喜んで眺めている。他愛ないものだ。

IOCSのNo.16$_H$，No.18$_H$は，文字を表示する位置をアドレスでしか指定できない。つまり，8ドット単位でしか文字表示位置を指示できないサービスである。16，24ドットの文字を扱う場合にはいいのだが，12ドットの文字を扱う場合にはこれでは文字を連続して表示することはできない。これを考慮したサービスも用意されているのだが，いずれ，またの機会にお話しすることとしよう。

図7　CGROMを利用した文字表示

```
-a a70a0
  000A70A0      ori.b    #$00,D0
                move.w   #'A',d1
  000A70A4      ori.b    #$00,D0
                move.l   #8,d2
  000A70AA      ori.b    #$00,D0
                move.l   #$16,d0
  000A70B0      ori.b    #$00,D0
                trap     #15        → ここまでは図4と同じ
  000A70B2      ori.b    #$00,D0
                move.l   #127,d3    → テキストVRAMの横バイト数－1をD3.lにセットし
  000A70B8      ori.b    #$00,D0
                sub.w    d1,d3      → d3.wからd1.wを引く
  000A70BA      ori.b    #$00,D0
                movea.l  d0,a1      → A1.lに表示するデータの入ったアドレスを入れ
  000A70BC      ori.b    #$00,D0
                movea.l  #$e40000,a2  → A2.lには表示するアドレスを入れる
  000A70C2      ori.b    #$00,D0
                move.l   #$18,d0    → D0.lにサービス番号18Hをセット
  000A70C8      ori.b    #$00,D0
                trap     #15        → IOCSを利用
  000A70CA      ori.b    #$00,D0

-b0  a70ca                          → ブレークポイントを設定して
-g=a70a0                            → プログラムを実行

        break at 000A70CA
PC=000A70CA USP=000857E4 SSP=000067F2 SR=0000 X:0  N:0  Z:0  V:0  C:0
D  0000FFFF 00080000 0000FFFF 0000007F  00000000 00000000 00000000 00000000
A  00000000 00F3AC20 00E40800 00000000  00000000 00000000 00000000 000857E4
ori.b   #$00,D0
-f e40000 e5ffff 0                  → 終了するときはマウスプレーンをクリアすること
-q
```

コラム：御仁のお言葉

moveとmoveaの違い

moveとmoveaの大きな違いはD0～D7，A0～A7のどちらのレジスタにデータをセットするかですが，それだけではありません。まず，moveaではバイトデータをセットすることができません。「movea.b」という命令はないのです。また，ワードデータをセットする場合の動作にも違いがあります。これは，負の数の表現方法と密接な関係があります。

1バイトのデータは00～FF$_H$を表現することができます。これは10進数でいえば0～255に相当しますが，これでは正の数しか扱えません。そこで次のような負の数を表すルールが作られました。

FF$_H$に1を加えれば100$_H$となりますが，繰り上がりを無視してバイトの範囲だけに注目すれば00$_H$と見なすことができます。そこでFF$_H$を，1加えれば0になる数，すなわち，－1と見なすことにしていたのです。そして00～7F$_H$は正の数（0～127），FF～80$_H$は負の数（－1～－128）とされました。同様にワードのデータでも，0000～7FFF$_H$は正の数，FFFF～8000$_H$は負の数となり，ロングワードのデータでも00000000～7FFFFFFF$_H$は正の数，FFFFFFFF～80000000$_H$は負の数としました。このような数の表現方法は2の補数表現と呼ばれます。同じFF$_H$というデータでも，正数だと見なせば255になりますが，2の補数表現だと見なせば－1になります。FF$_H$をどちらだと見なして扱うかはプログラマに任されています。

moveとmoveaの違いに話を戻しましょう。move命令は，

```
move.w   #$7FFF,d0  →  D0＝????7FFFH
move.w   #$8000,d0  →  D0＝????8000H
move.l   #$7FFF,d0  →  D0＝00007FFFH
move.l   #$8000,d0  →  D0＝00008000H
```

（?のところは変化しない）

というように指定されたデータを素直にセットしますが，movea命令は必ずデータをロングワードに変換してからセットします。これはアドレスレジスタが，本来アドレス（ロングワードデータ）を格納するものであることに起因しています。しかも変換時には，指定されたデータを2の補数表現だと見なして変換を行うのです。

```
movea.w  #$7FFF,a0  →  A0＝00007FFFH
movea.w  #$8000,a0  →  A0＝FFFF8000H
movea.l  #$7FFF,a0  →  A0＝00007FFFH
movea.l  #$8000,a0  →  A0＝00008000H
```

「movea.l」では，move.lのときと同様，7FFF$_H$も8000$_H$と上位に4桁の0を補ったロングワードデータとしてそのままセットされます。しかし，「movea.w」はデータを符号つき数と見なしてロングワードに変換します（符号拡張される，といいます）。ですから，7FFF$_H$は正の数32767（＝00007FFF$_H$），8000$_H$は負の数－32768（＝FFFF8000$_H$）としてセットされることになります。

バイトデータを扱えないことに加え，ワードデータをセットするときには符号拡張が起こりロングワードデータとしてセットされる，これがmoveaのmoveとの大きな違いです。

システムの安全性

御仁がコラムで取り上げたmove命令の実例,

move.b　$e00000,$e00001

は，今回図５や図６でやった方法では実行することができない。これはテキストVRAMがユーザープログラムから直接扱うことのできないスーパーバイザ領域にあるためだ。吾輩は，ユーザーの不用意なプログラムによってシステムの重要な部分が書き換えられてしまうことを防ぐため，スーパーバイザモード，ユーザーモードの２つのモードをもってシステムとユーザを切り離しているのである。

完全とはいえないが，これでかなりシステムの安全性は向上している。以前は「マシン語＝暴走」といわれるほどマシン語を扱うのは微妙な問題であった。プログラマがデータをセットするアドレスをうっかり間違えたがために，コンピュータはあらぬ命令を延々と実行し続け，挙げ句の果てに画面全体を点滅させたり，音を鳴らし続けたり，画面をサイケな絵で満たしたりしたものである。不幸にしてディスクにデータを書き込むルーチンが不正に使用されたため，大切なデータを書き込んだディスクをオシャカにされたユーザーもいる。なにを隠そう，御仁もその１人である。この状態から抜け出すためには，リセットスイッチを使ってシステムの起動からやり直すしかない。なにが悪かったのかは，命令を１つひとつ自分で追いかけていき，いったいなにが起きたのかを解析するしかなかった。

図８はあえて先のプログラムを実行してみた例である。吾輩はユーザープログラムが不正にスーパーバイザ領域を使おうとする場合には，このように強制的にプログラムの実行を中断してしまう。

奇数アドレスからワード・ロングワードデータを取り出そうなどとした場合も同様である。

こうすることによって吾輩は我身の安全を図ることができ，ユーザーは自分のプログラムのなにが悪かったのかを知ることができる（少なくとも糸口にはなる）というわけである。

といったところで今回はお別れである。次回をお楽しみに。

図8　スーパーバイザ領域を扱ってみる

```
-a a70a0
  000A70A0      ori.b   #$00,D0
                move.b  $e00000,$e00001 → E00000HをE00001Hにコピー
  000A70AA      ori.b   #$00,D0
                .
-b0  a70aa                              → ブレークポイントを設定して
-g=a70a0                                → 実行

Exceptional Abort By bus error          → エラーが起きたので中断される。理由は
 By Memory Access of 00E00000           → E00000Hのデータを扱おうとしたから
 at   000A70A0  move.b  $00E00000,$00E00001 → この命令で中断された

system status =  I/N=I R/W=R FC=1
PC=000A70A0 USP=000857E4 SSP=000067F2 SR=8000 X:0  N:0  Z:0  V:0  C:0
D  00000000 00000000 00000000 00000000  00000000 00000000 00000000 00000000
A  00000000 00000000 00000000 00000000  00000000 00000000 00000000 000857E4
move.b  $00E00000,$00E00001
```

わが頭脳MC68000

MC68000は吾輩の動作の中核を成すLSIである。このLSIはメモリからデータを読み，それをプログラムだと見なしてさまざまな動作を行う。その最も基本的な動作が「データの移動」である。メモリの$E00000_H$に入っているデータ，すなわち，画面左上隅に表示されているデータを$E00001_H$に「移動」すれば，左上隅８ドットと同じ模様がその右隣８ドットに表示されることになる。$E00000_H$というのは，前回もやったようにテキストプレーン0の左上隅８ドットに対応するデータが入っているアドレスである。

「移動」とわざわざカッコを付けてあるのには理由がある。MC68000にとっての「移動」は，人間の言葉で言う「複写」に相当する。$E00000_H$に入っているデータを$E00001_H$に移動するとは，$E00000_H$のデータのコピーをとり，それを$E00001_H$にセットすることにほかならない。

●データの「移動」

データの「移動」は，moveという命令で行う。$E00000_H$に入っているデータを$E00001_H$に移動するには，

move.b $e00000, $e00001

となる。moveの後ろに「.b」がついているのは，１バイトのデータを移動するという印である。１ワードのデータを移動するなら「.w」，１ロングワードのデータを移動するなら「.l」をつければいい。ちなみに「.w」は省略可能である。

MC68000は自分の内部にも若干のメモリを持っている。数は少ないがこれは超高速なメモリでありレジスタと呼ばれている。レジスタにはD0〜D7，A0〜A7の16個がある。D0〜D7はデータレジスタと呼ばれており，A0〜A7はアドレスレジスタと呼ばれている。いずれも１ロングワードのデータを入れることができ，これらのレジスタにデータを「移動」するのにもmove命令が使われる。

move.b　$e00000, d0

なら$E00000_H$に入っている１バイトのデータがD0にコピーされるし，

move.l　$e00000, d1

なら$E00000_H$，$E00001_H$，$E00002_H$，$E00003_H$に入っている４つの１バイトデータ，すなわち１ロングワードデータがD0にコピーされる。ここで注意しておきたいのは，MC68000は奇数アドレスから始まるワード・ロングワードデータを扱うことはできないという点である。したがって，

move.w　$e00001, $e00002
move.l　d0, $e00001

などという命令はエラーとなる。

メモリに入っているデータではなく，レジスタに直接データをセットしたい場合もあろう。このときには，

move.l　#データ, レジスタ

のようにmove命令を使うことになっている。直接セットするデータに「#」がつけてあることに注目していただきたい。

●計算機能

MC68000は単純な計算も行うことができる。

add　データ, データレジスタ

はデータとデータレジスタの加算を行い，結果をデータレジスタに格納するし，

sub　データ, データレジスタ

はデータレジスタからデータを引き，結果をデータレジスタに格納する。

add，subにも「.b」「.w」「.l」を付けることができるが，この場合には指定された範囲だけが計算の対象となる。

move.l　#$ff, d0 → D0＝000000FF
add.b　#1, d0　→ D0＝00000000

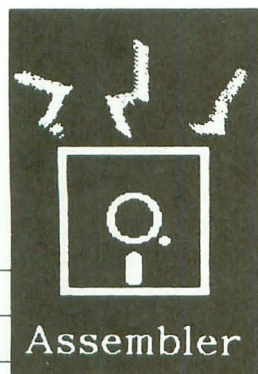
FF_Hに1を加えると答えは100_Hだが，加算をバイトで行っているため，答えのバイト部分だけが繰り上がりを無視してD0のバイト部分（下２桁）にセットされるのである。

move.l　#0, d0
sub.w　#1, d0

ならどうなるかなどいろいろ考えてみていただきたい。

続・必須のラインルーチン

Murata Toshiyuki 村田 敏幸

グラフィック描画の必需品ということで取り上げたラインルーチンですが，先月紹介したものでは納得がいかないと村田氏の気が変わってしまいました。というわけで，予定を変更してのラインルーチン続編です。これもプログラミングにはよくあることですが……。

今回は（も）予定を変更して，前回やったライン描画を若干補足する。気紛れで，もう少し速くしてやりたくなったのだ。先月のリスト6からリスト9への改良は，ある程度のシェイプアップに加え，比較的よく使われる水平/垂直（ついでに45度）の線分を専用ルーチンで処理することにより実用上の性能を稼ぐ，というだけのことだった。それはそれで効果はあるとはいえ，どうも中途半端な感じがある。ごく簡単な工夫をしていないのだ。今月は先月のラインルーチンにその工夫を加え，さっと切り上げる。

サブルーチンの評価

まず，IOCSコールLINEと先月作ったサブルーチンglineの性能を把握しておこう。リスト1のようなプログラムで512×512ドットの画面に50000本のランダムな線分を描かせて，IOCSコールONTIMEにより1/100秒単位で計時してみた。「引数のセットはするが実際には線分描画を行わないダミーのループ」と，「実際に線分を描くループ」との差をとることで，ライン描画以外の余計な処理にかかる時間を除外している。表示される時間は純粋にラインルーチン呼び出し50000回の合計時間と見てよい。ただし，乱数により，描く線分の長さは予想以上に大きく変動するので，小数点以下にはかなりの誤差があると考えること。

リスト1は1回に1種類のラインルーチンの実行時間を測定するだけだから，先月の2本のglineそれぞれについて計時するためには，リンク時にどちらか適切なモジュールを選んでリンクする必要がある。また，IOCSコールLINEの実行時間を計るときには，リスト1中，77～79行を削り，代わって80行を復活することで対処する。さらに，クリッピングルーチンの性能を加味したければ，8行と32～34行を復活すると，クリッピングウィンドウが(64,64)-(447,447)に設定される。

プログラミングのうえでは，リスト1には特に見るべき点はないと思う。1カ所だけ，FLOATn.Xのファンクションを利用して（手抜き）得た0～32767の疑似乱数を0～511の座標値の範囲に収めるために，

```
lsr.w   #6,d0
```

を使っている点を指摘しておこう（88行）。FLOATn.Xで使用している疑似乱数発生アルゴリズムの性格で，

```
andi.w  #511,d0
```

のように下位ビットを利用するよりは，シフトして上位ビットを利用したほうが概してばらつきのよい乱数が得られるのだった[1]。

OPMDRV.Xを外した状態で測定した結果を以下に示す（おっと，クロックは10MHz。単位は秒）。

gline（修正版）	55.1
gline（初期版）	62.98
IOCS（IOCS.X版）	96.70
IOCS（ROM版）	197.87

見てのとおり，先月作ったサブルーチンglineは，IOCS.Xを組み込んだときのIOCSコールより速いという結果が出た。ただし，この結果を鵜のみにしてはいけない。世の多くのベンチマークテストがそうであるように，このテストは作為的であり，正直な読者を欺くようにできている。

第1に，ルーチンの性格の違いがある。IOCSのLINEは毎回の呼び出しごとにグラフィック画面が初期化されているかどうか確認するし，描画色が画面モードに合った範囲に収まっているかどうかも調べるし，IOCSコールの呼び出し手順そのもののオーバーヘッドもある。対して，サブルーチンglineは画面モード周りのエラーチェックをしていない。

それに加えて，機能の差がある。IOCSのLINEはラインスタイルをサポートしているが，glineは実線専用だ。試算してみたところ，ラインスタイルをサポートすると実線専用のルーチンと比べて60～70%速度が低下すると考えられ，その点を割り引けばIOCS.X版LINEの潜在的な性能は先月作ったラインルーチン（素直に作った版）とトントンだといえる[2]。ちなみに，リスト2は先月のリスト9をラインスタイル対応にしたものだ。追加・修正点は左側に“＋”記号をつけることで示してある。要はd7.wに16ビットのビットパターンを入れておき，これを

1) これはX-BASICのrand( )関数にも当てはまる。

2) IOCSもラインスタイルがFFFFH（実線）のときには専用ルーチンで対応するぐらいのことをしてくれたってバチは当たらないようにも思うが。

3) rolでビットパターンを回転して左側からはみ出したビットは，最下位ビットと同時にCフラグに格納される。

4) 四捨五入のもつ非対称性を考えよう。

5) (a XOR b) XOR b = aだ。

rol.wで回転させて，はみ出したビットの1/0に応じて点を打つか打たないかを決めることでラインスタイルを実現している[3]。実行時間を測定してみると，IOCS.X版LINEと同程度にまで速度が低下しているのがわかる。

再びラインの高速化

では，改めてラインルーチンの高速化を目指そう，すでに先月のリスト9のループはたぶん限界までぜい肉を削ってあり，これ以上速くしようと思ったら，アルゴリズムを練り直すか，場合分けを推し進めて特定パターンに応じた専用ルーチンを用意するかしかないように見える。しかし，線分は基本的に上下左右対称なのだから，両端から同時に描けば誤差項の計算回数が半分ですみ，ループ回数も半減できるだろう。

先月のリスト9を改造すると，リスト3のようになった。修正箇所はやはり“+”で示してある。

主な修正点は，a2に終点のG-RAMアドレスを求めるようになったこと(36～40行)，両端の2点を並行して処理するように74～82行や100～116行のループを修正したこと，それに合わせ，ループカウンタを半減したことだ(69～70行，94～95行)。ループカウンタを半減する部分では，線分が奇数ピクセルだった場合に備えて，微妙なつじつま合わせをしている。d4.bは線分の長さが奇数ピクセルかどうかを覚えておくフラグとして使われており，ループから抜けたあと，まだ中央の1ピクセルが残っているかどうかの判断材料になる。

```
scs.b  d4
```

は条件セット命令で，Cフラグが1ならd4.bをFF_Hにし，Cフラグが0ならd4.bを00_Hにすることを思い出してほしい。

ところで，半端になった中央のピクセルを特別扱いするのをやめれば，プログラムはもう少しすっきりする。69行や94行で1引いている部分を削り，ループカウンタを1つ大きくして，中央のピクセルだけはダブって描画するわけだ。そうすると，d4.bをフラグにして云々というあたりの余計な処理はみんないらなくなるから，多少は実行速度も上がるかもしれない。しかし，その場合，線分の傾きによっては中央だけ線分が太くなってしまうという副作用が見られ[4]，また，このラインルーチンを改造してXORモード対応にしたときにも問題が生じる[5]。

さて，終点側から点を打っていく処理は，始点側から点を打つこれまでの処理とポインタの移動方向を反対にすれば実現できる。この周辺では例によって細部を微調整して処理効率を稼いでいる。傾きが緩やかな線分を描く74行以下のループでは，75行でポインタをプリデクリメントできるよう，あらかじめ72行でa2に2を足してあるのがそれだ。また，急な線分を描く100行以下のループでは，本来a2を「ポストデクリメント」してからd5.wを引くところを，d7.wにd5.w+2を先に求めておくことで113行のように簡略化している。すぐ上の110～111行は，

```
move.w  d0,(a0)
adda.w  d7,a0
```

と書いても同じ意味になるから，これと見比べれば112～113行の意味は明確になると思う。

実行時間を測定してみると,50000本のライン描画に41.08秒という結果になった。だいたい1200ライン/秒だ。この数字は一般的に見て速いほうなのかどうかはよくわからないが，少なくとも先月の版よりは大きく改善されているし，まあ，満足できる性能といえるだろう。

垂直線分描画ルーチンの高速化

ついでだから，先月の最後にヒントだけを示した垂直線分描画ルーチンの高速化の実例も示しておく。リスト4だ。リスト3の142行以下と差し替えて使うようにできている。

ご覧のように，ループ展開っぽい姑息な技であり，あまり解説したくはない。ディスプレースメントつきアドレスレジスタ間接形式適用のmove命令をずらっと32個(実画面512ドットモード時。1024ドットモード時は16個) 並べ，それをループで括ってあるのが何を意味するかは考えるまでもあるまい。ループ1回につき長さ32 (あるいは16) の垂直線分を描くわけだ。それを描くべき線分の長さを32で割った回数繰り返す。32で割った余りは,「ループの途中に飛び込む」ことで描く。

＊

というわけで，今月は本当にさっさと終わる。来月は，今度こそ多角形の塗り潰しを取り上げる。

リスト1 GLINETEST.S

```
 1: *           glineのテスト用プログラム
 2: *
 3:             .include        doscall.mac
 4:             .include        iocscall.mac
 5:             .include        const.h
 6: *
 7:             .xref   gline
 8: *           .xref   setcliprect
 9: *
10: FPACK       macro   callno
11:             .dc.w   callno
12:             endm
13: *
14: __RAND              equ     $fe0e   *乱数(0~32767)
15: __IUSING            equ     $fe18   *整数→文字列変換
16:                                     *(桁数指定つき)
17: IOCS_GL3            equ     12
18: *
19:             .text
20:             .even
21: *
22: ent:
23:             lea.l   inisp(pc),sp
24:
25:             moveq.l #IOCS_GL3,d1        *画面を512x512
26:             IOCS    _CRTMOD             * 65536色モード
27:             IOCS    _G_CLR_ON           *グラフィックON
28:
29:             suba.l  a1,a1               *スーパーバイザ
30:             IOCS    _B_SUPER            * モードへ
31:
32: *           pea.l   window(pc)          *クリッピング
33: *           jsr     setwindow           * ウィンドウを
34: *           addq.l  #4,sp               * 設定する
35:
36:             lea.l   argbuf(pc),a1       *a1 = 引数受け渡し領域
```

```
 37:
 38:          IOCS     _ONTIME             *ループなどにかかる
 39:          move.l   d0,-(sp)            *  余分な時間を
 40:          bsr      test1               *  計っておく
 41:          IOCS     _ONTIME             *
 42:          sub.l    (sp)+,d0            *
 43:          bpl      tskip1              *
 44:          addi.l   #24*3600*100,d0     *
 45: tskip1:  move.l   d0,-(sp)            *
 46:
 47:          IOCS     _ONTIME             *描画にかかる
 48:          move.l   d0,-(sp)            *  時間を計る
 49:          bsr      test2               *
 50:          IOCS     _ONTIME             *
 51:          sub.l    (sp)+,d0            *
 52:          bpl      tskip2              *
 53:          addi.l   #24*3600*100,d0     *
 54: tskip2:  sub.l    (sp)+,d0            *d0 = 正味の描画時間
 55:
 56:          lea.l    temp(pc),a0         *時間を
 57:          moveq.l  #7,d1               *  10進7桁右詰めで
 58:          FPACK    __IUSING            *  文字列に変換する
 59:
 60:          bsr      conv                *1/100秒単位から秒単位へ
 61:          pea.l    temp(pc)            *描画に要した
 62:          DOS      _PRINT              *  時間を表示する
 63:          pea.l    secmes(pc)          *
 64:          DOS      _PRINT              *
 65:
 66:          DOS      _EXIT
 67: *
 68: test1:
 69:          move.w   #50000-1,d7
 70: loop1:   bsr      setarg
 71:          dbra     d7,loop1
 72:          rts
 73: *
 74: test2:
 75:          move.w   #50000-1,d7
 76: loop2:   bsr      setarg
 77:          pea.l    (a1)
 78:          jsr      gline
 79:          addq.l   #4,sp
 80: *        IOCS     _LINE
 81:          dbra     d7,loop2
 82:          rts
 83: *
 84: setarg:
 85:          movea.l  a1,a0
 86:          move.w   #4-1,d6
 87: arglp:   FPACK    __RAND
 88:          lsr.w    #6,d0
 89: *        lsr.w    #5,d0
 90: *        subi.w   #256,d0
 91:          move.w   d0,(a0)+
 92:          dbra     d6,arglp
 93:          rts
 94: *
 95: conv:
 96:          lea.l    temp+7(pc),a1
 97:          lea.l    temp+8(pc),a2
 98:          moveq.l  #$20,d1
 99:          moveq.l  #'0',d2
100:          clr.b    (a2)
101:          move.b   -(a1),-(a2)
102:          move.b   -(a1),d0
103:          cmp.b    d1,d0
104:          bne      skip1
105:          move.b   d2,d0
106: skip1:   move.b   d0,-(a2)
107:          move.b   #'.',-(a2)
108:          move.b   -(a2),d0
109:          cmp.b    d1,d0
110:          bne      skip2
111:          move.b   d2,(a2)
112: skip2:   rts
113: *
114:          .data
115:          .even
116: *
117: window:  .dc.w    64,64,511-64,511-64
118: argbuf:  .dc.w    0,0,0,0,63,$ffff
119: secmes:  .dc.b    ' sec.',CR,LF,0
120: *
121:          .bss
122:          .even
123: *
124: temp:    .ds.b    10                  *数値→文字列変換用
125: *
126:          .stack
127:          .even
128: *
129:          .ds.l    1024
130: inisp:
131:          .end     ent
```

リスト2　GLINE.S(ラインスタイル対応版)

```
  1: *        線分描画(ラインスタイル対応)
  2: *
  3:          .include          gconst.h
  4:          .include          gmacro.h
  5: *
  6:          .xdef    gline
  7:          .xref    gramadr
  8:          .xref    glclip
  9: *
 10:          .offset  0
 11: *
 12: X0:      .ds.w    1
 13: Y0:      .ds.w    1
 14: X1:      .ds.w    1
 15: Y1:      .ds.w    1
 16: COL:     .ds.w    1
 17: BITPAT:  .ds.w    1
 18: *
 19:          .text
 20:          .even
 21: *
 22: gline:
 23: ARGPTR = 8
 24:          link     a6,#0
 25:          movem.l  d0-d7/a0-a1,-(sp)
 26:
 27:          move.l   ARGPTR(a6),a1     *a1 = 引数受け渡し領域
 28:          move.w   BITPAT(a1),d7     *d7 = ラインスタイル
 29:          beq      done
 30:
 31:          movem.w  (a1),d0-d3        *d0-d3に座標を取り出す
 32:
 33:          jsr      glclip            *クリッピングする
 34:          bne      done              *Z=0なら描画の必要なし
 35:
 36:          jsr      gramadr           *始点のG-RAM上アドレスを得る
 37:
 38:          sub.w    d0,d2             *d2 = x1-x0
 39:          move.w   d2,d4             *d4 = d2
 40:          ABS      d2                *d2 = dx = abs(x1-x0)
 41:          SGN      d4                *d4 = sx = sgn(x1-x0)
 42:
 43:          sub.w    d1,d3             *d3 = y1-y0
 44:          move.w   d3,d5             *d5 = d3
 45:          ABS      d3                *d3 = dy = abs(y1-y0)
 46:          SGN      d5                *d5 = sy = sgn(y1-y0)
 47:
 48:          add.w    d4,d4             *d4 = sx*2
 49:          moveq.l  #GSFTCTR,d0       *
 50:          asl.w    d0,d5             *d5 = sy*1024 (or 2048)
 51:
 52:          move.w   COL(a1),d0        *d0 = color
 53:
 54:          cmp.w    d3,d2             *dy > dxならば
 55:          bcs      yline             *  yについてループ
 56:
 57:                                     *dx >= dyのとき
 58: xline:   move.w   d2,d1             *d1 = d2
 59:          neg.w    d1                *d1 = e = -dx
 60:          move.w   d2,d6             *d6 = n = dx
 61:          add.w    d2,d2             *d2 = dx*2
 62:          add.w    d3,d3             *d3 = dy*2
 63:                                     *do {
 64: xline0:  rol.w    #1,d7             *
 65:          bcc      xskip             *
 66:          move.w   d0,(a0)           *   pset(x,y)
 67: xskip:   adda.w   d4,a0             *   x += sx
 68:          add.w    d3,d1             *   e += 2*dy
 69:          bmi      xline1            *   if (e >= 0) {
 70:          adda.w   d5,a0             *     y += sy
 71:          sub.w    d2,d1             *     e -= 2*dx
 72:                                     *   }
 73: xline1:  dbra     d6,xline0         *} while (--n >= 0)
 74:          bra      done
 75:                                     *dx < dyのとき
 76: yline:   move.w   d3,d1             *d1 = d3
 77:          neg.w    d1                *d1 = e = -dy
 78:          move.w   d3,d6             *d6 = n = dy
 79:          add.w    d2,d2             *d2 = dx*2
 80:          add.w    d3,d3             *d3 = dy*2
 81:                                     *do {
 82: yline0:  rol.w    #1,d7             *
 83:          bcc      yskip             *
 84:          move.w   d0,(a0)           *   pset(x,y)
 85: yskip:   adda.w   d5,a0             *   y += sy
 86:          add.w    d2,d1             *   e += 2*dx
 87:          bmi      yline1            *   if (e >= 0) {
 88:          adda.w   d4,a0             *     x += sx
 89:          sub.w    d3,d1             *     e -= 2*dy
 90:                                     *   }
 91: yline1:  dbra     d6,yline0         *} while (--n >= 0)
 92: done:    movem.l  (sp)+,d0-d7/a0-a1
 93:          unlk     a6
 94:          rts
 95:
 96:          .end
```

リスト3　GLINE.S(高速化版)

```
  1: *        線分描画
  2: *
  3:          .include          gconst.h
  4: *
  5:          .xdef    gline
  6:          .xref    gramadr
  7:          .xref    glclip
  8: *
```

```
  9:          .offset 0
 10: *
 11: X0:      .ds.w   1
 12: Y0:      .ds.w   1
 13: X1:      .ds.w   1
 14: Y1:      .ds.w   1
 15: COL:     .ds.w   1
 16: *
 17:          .text
 18:          .even
 19: *
 20: gline:
 21: ARGPTR = 8
 22:          link    a6,#0
 23:          movem.l d0-d7/a0-a2,-(sp)
 24:
 25:          move.l  ARGPTR(a6),a1      *a1=引数受け渡し領域
 26:          movem.w (a1),d0-d3         *d0-d3に座標を取り出す
 27:
 28:          jsr     glclip             *クリッピングする
 29:          bne     done               *Z=0なら完全不可視
 30:
 31:          cmp.w   d2,d0              *x0 >= x1を保証する
 32:          bge     gline0             *
 33:          exg.l   d0,d2              *
 34:          exg.l   d1,d3              *
 35:
 36: gline0:  jsr     gramadr            *終点のG-RAMアドレスを得る
 37:          movea.l a0,a2              *a2 = 終点のG-RAMアドレス
 38:          exg.l   d0,d2              *
 39:          exg.l   d1,d3              *この時点でx0 <= x1
 40:          jsr     gramadr            *始点のG-RAMアドレスを得る
 41:                                     *a0 = 始点のG-RAMアドレス
 42:
 43:          move.w  #GNBYTE,d5         *d5 = 横1ライン分のバイト数
 44:          sub.w   d1,d3              *d3 = y1-y0
 45:          beq     hor_line           *y0 = y1 なら水平線
 46:          bpl     gline1
 47:          neg.w   d3
 48:          neg.w   d5
 49: gline1:  sub.w   d0,d2              *d2 = x1-x0 ( >=0 )
 50:          beq     ver_line           *x0 = x1 なら垂直線
 51: *この時点で
 52: *        d2 = dx = abs(x1-x0) ( > 0 )
 53: *        d3 = dy = abs(y1-y0) ( > 0 )
 54: *        d5 = sy = sgn(y1-y0) ( -1 or 1 )
 55: *        (ただしd5はGNBYTE倍済み)
 56:
 57:          move.w  COL(a1),d0         *d0 = color
 58:
 59:          cmp.w   d3,d2              *dy > dxならば
 60:          bcs     yline              * yについてループ
 61:          beq     xyline             *dy = dxならば45度の線
 62:
 63:                             *dx >= dyのとき
 64: xline:   move.w  d2,d1              *d1 = dx
 65:          neg.w   d1                 *d1 = e = -dx
 66:          move.w  d2,d6              *d6 = n = dx
 67:          add.w   d2,d2              *d2 = dx*2
 68:          add.w   d3,d3              *d3 = dy*2
 69:          subq.w  #1,d6              *dbraのことを計算に入れて
 70:          lsr.w   #1,d6              * ループカウンタを半減
 71:          scs.b   d4                 *奇数ピクセルのとき非0
 72:          addq.l  #2,a2              *プリデクリメントする分補正
 73:                                     *do {
 74: xline0:  move.w  d0,(a0)+           *   pset(x++,y)
 75:          move.w  d0,-(a2)           *   pset(--x',y')
 76:          add.w   d3,d1              *   e += 2*dy
 77:          bmi     xline1             *   if (e >= 0) {
 78:          adda.w  d5,a0              *     y += sy
 79:          suba.w  d5,a2              *     y'-= sy
 80:          sub.w   d2,d1              *     e -= 2*dx
 81:                                     *   }
 82: xline1:  dbra    d6,xline0          *} while (--n >= 0)
 83:
 84:          tst.b   d4                 *奇数ピクセル?
 85:          beq     done               * そうじゃない
 86:          bra     odd                *中央のピクセルを点灯
 87:
 88:                             *dx < dyのとき
 89: yline:   move.w  d3,d1              *d1 = dy
 90:          neg.w   d1                 *d1 = e = -dy
 91:          move.w  d3,d6              *d6 = n = dy
 92:          add.w   d2,d2              *d2 = dx*2
 93:          add.w   d3,d3              *d3 = dy*2
 94:          subq.w  #1,d6              *dbraのことを計算に入れて
 95:          lsr.w   #1,d6              * ループカウンタを半減
 96:          scs.b   d4                 *奇数ピクセルのとき非0
 97:          move.w  d5,d7              *d7 = d5 + 2
 98:          addq.w  #2,d7              *
 99:                                     *do {
100: yline0:  add.w   d2,d1              *  e += 2*dx
101:          bpl     yline1             *  if (e < 0) {
102:          move.w  d0,(a0)            *    pset(x,y)
103:          adda.w  d5,a0              *    y += sy
104:          move.w  d0,(a2)            *    pset(x',y')
105:          suba.w  d5,a2              *    y'-= sy
106:                                     *  }
107:          dbra    d6,yline0
108:          bra     done0
109:                                     *  else {
110: yline1:  move.w  d0,(a0)+           *    pset(x++,y)
111:          adda.w  d5,a0              *    y += sy
112:          move.w  d0,(a2)            *    pset(x',y')
113:          suba.w  d7,a2              *    x'--, y'-= sy
114:          sub.w   d3,d1              *    e -= 2*dy
115:                                     *  }
116:          dbra    d6,yline0          *} while (--n >= 0)
117:
118: done0:   tst.b   d4                 *奇数ピクセル?
119:          beq     done               * そうじゃない
120: odd:     move.w  d0,(a0)            *中央のピクセルを点灯
121:
122: done:    movem.l (sp)+,d0-d7/a0-a2
123:          unlk    a6
124:          rts
125:
126: hor_line:                   *水平線
127:          sub.w   d0,d2              *d2 = dx = x1-x0
128:          move.w  COL(a1),d1         *d1 = color
129:          move.w  d1,d0              *d0 = color
130:          swap.w  d0                 *
131:          move.w  d1,d0              *d0.l = color_color
132:          addq.w  #1,d2              *d2 = dx+1 = ピクセル数
133:          bclr.l  #0,d2              *
134:          beq     hskip              *
135:          move.w  d0,(a0)+           *奇数ピクセルの分
136: hskip:   lea.l   hline(pc),a1
137:          suba.w  d2,a1
138:          jmp     (a1)
139:          .dcb.w  GNPIXEL/2,$20c0    *move.l d0,(a0)+
140: hline:   bra     done
141:
142: xyline:                     *45度の線
143:          addq.w  #2,d5              *d5 = 2±GNBYTE
144: ver_line:                   *垂直線
145:          move.w  COL(a1),d0         *d0 = color
146: vloop:   move.w  d0,(a0)            *pset(x,y)
147:          adda.w  d5,a0              *y += sx
148:          dbra    d3,vloop           *dy+1回繰り返す
149:          bra     done
150:
151:          .end
```

リスト4 GLINEV.S

```
  1: xyline:                    *45度の線
  2:         move.w  COL(a1),d0         *d0 = color
  3: xyloop: move.w  d0,(a0)+           *pset(x++,y)
  4:         adda.w  d5,a0              *y += sx
  5:         dbra    d3,xyloop          *dy+1回繰り返す
  6:         bra     done
  7:
  8: ver_line:                  *垂直線
  9:         tst.w   d5                 *d5 = sy
 10:         bpl     vskip              *a0 <= a2を
 11:         movea.l a2,a0              * 保証する
 12: vskip:  move.w  d3,d1
 13:         move.l  #$0000_8000,d5     *long!
 14: .ifdef  _1024
 15:         andi.w  #16-1,d1           *d1 = dy%16
 16:         lsr.w   #4,d3              *d3 = dy/16
 17: .else
 18:         andi.w  #32-1,d1           *d1 = dy%32
 19:         lsr.w   #5,d3              *d3 = dy/32
 20: .endif
 21:         move.w  d1,d2
 22:         moveq.l #GSFTCTR,d0
 23:         lsl.w   d0,d2
 24:         adda.w  d2,a0
 25:         move.w  COL(a1),d0         *d0 = color
 26:         addq.w  #1,d1
 27:         lsl.w   #2,d1
 28:         neg.w   d1
 29:         lea.l   vnext(pc),a1
 30:         jmp     0(a1,d1.w)
 31: .ifdef  _1024
 32: vloop:  move.w  d0,-$7800(a0)      *長さ16の
 33:         move.w  d0,-$7000(a0)      * 垂直線分を描く
 34:         move.w  d0,-$6800(a0)
 35:         move.w  d0,-$6000(a0)
 36:         move.w  d0,-$5800(a0)
 37:         move.w  d0,-$5000(a0)
 38:         move.w  d0,-$4800(a0)
 39:         move.w  d0,-$4000(a0)
 40:         move.w  d0,-$3800(a0)
 41:         move.w  d0,-$3000(a0)
 42:         move.w  d0,-$2800(a0)
 43:         move.w  d0,-$2000(a0)
 44:         move.w  d0,-$1800(a0)
 45:         move.w  d0,-$1000(a0)
 46:         move.w  d0,-$0800(a0)
 47:         move.w  d0,(a0)
 48:         adda.l  d5,a0
 49: .else
 50: vloop:  move.w  d0,-$7c00(a0)      *長さ32の
 51:         move.w  d0,-$7800(a0)      * 垂直線分を描く
 52:         move.w  d0,-$7400(a0)
 53:         move.w  d0,-$7000(a0)
 54:         move.w  d0,-$6c00(a0)
 55:         move.w  d0,-$6800(a0)
 56:         move.w  d0,-$6400(a0)
 57:         move.w  d0,-$6000(a0)
 58:         move.w  d0,-$5c00(a0)
 59:         move.w  d0,-$5800(a0)
 60:         move.w  d0,-$5400(a0)
 61:         move.w  d0,-$5000(a0)
 62:         move.w  d0,-$4c00(a0)
 63:         move.w  d0,-$4800(a0)
 64:         move.w  d0,-$4400(a0)
 65:         move.w  d0,-$4000(a0)
 66:         move.w  d0,-$3c00(a0)
 67:         move.w  d0,-$3800(a0)
 68:         move.w  d0,-$3400(a0)
 69:         move.w  d0,-$3000(a0)
 70:         move.w  d0,-$2c00(a0)
 71:         move.w  d0,-$2800(a0)
 72:         move.w  d0,-$2400(a0)
 73:         move.w  d0,-$2000(a0)
 74:         move.w  d0,-$1c00(a0)
 75:         move.w  d0,-$1800(a0)
 76:         move.w  d0,-$1400(a0)
 77:         move.w  d0,-$1000(a0)
 78:         move.w  d0,-$0c00(a0)
 79:         move.w  d0,-$0800(a0)
 80:         move.w  d0,-$0400(a0)
 81:         move.w  d0,(a0)
 82:         adda.l  d5,a0
 83: .endif
 84: vnext:  dbra    d3,vloop
 85:         bra     done
 86:
 87:         .end
```

[第8回] 関数って何だろう(その2)

Nakamori Akira
中森 章

C言語の格子ということで取り上げた関数ですが，今月はその2回目。プログラムのモジュール化のためにぜひとも必要なグローバル変換，ローカル変数，引数などの理解を深めておきましょう。また，再帰呼び出しについても解説します。

最近，めっきりと小説を読まなくなってしまいましたが，コバルト文庫の「星子ひとり旅シリーズ」だけは毎回欠かさずチェックし，ときおり過ぎ去りし青春時代を回顧している中森章です。

さて，今回も前回の続きで関数についての話をします。前回は関数の定義方法を中心に説明をしました。しかし，関数を定義できるようになっても，グローバル変数，ローカル変数，引数といった概念を押さえておかなければ，関数の第一目標であるモジュール化を効率よく行えるようになりません。そこで，今回はこの不足している部分の知識を補うとともに，関数定義の特殊（でもないけど）な例である再帰呼び出しの方法について解説することにしましょう。

グローバル変数とローカル変数

これまでプログラムの中で変数をいろいろと使用してきました。その変数を宣言する部分は関数の外であったり中であったりしたわけですが，それらをそれほど区別をして使い分けていたわけではありません。しかし，関数の外で宣言する変数と関数の中で宣言する変数の間には大きな違いがあります。関数について学んだついでにこれらの変数の違いについて学んでおきましょう。

前回も述べましたが，プログラムで関数を用いることの意義のひとつはモジュール化です。モジュール化された関数は，それを呼び出す側からはブラックボックスとして扱われます。つまり，関数呼び出しのインタフェイス（引数の渡し方）を守ってさえいれば，その関数の提供する機能がどのような方法で実現されているかは呼び出す側の知ったことではありません。そのため，モジュール化された関数やそれを呼び出すプログラム（関数）はお互いに干渉しあわないことが必要になります。

もし，モジュール化された関数とそれを呼び出すプログラムの間で実体が異なる変数は同じ名前にしてはいけないという制限があるなら，お互いに名前がぶつからないように注意深く変数名を決めなくてはなりません。

このときに必要になるのは「変数表」という変数名とその変数の内容の一覧表ですが，どの関数からも共通に参照される変数ならともかく，ある関数内でしか使われない一時的な変数までも「変数表」に載せているとその表の変数の個数が爆発してしまいます。それに，変数がどういう目的で使用されるか変数名でそれなりに推測でき，かつ一意な変数名を考えるのにも相当な努力が必要です。巨大なプログラムをわかりやすく作るというのがモジュール化の利点ですが，モジュール化のためにこんな苦労をしていたのではその意味がなくなってしまうでしょう。

そこで，C言語などの高級言語[1]では，変数にはある関数によってしか参照できない変数とどの関数からも参照できる変数の2種類が用意されています。ある関数によってしか参照できない変数はローカル（局所的）変数と呼ばれます。一方，どの関数からも参照できる変数はグローバル（大域的）変数と呼ばれます。ローカル変数はひとつの関数に固有な変数で，関数の外側(呼び出し側)からは決して参照することができません。当然，異なる関数の間でなら同じ名前のローカル変数を使用してもなんの問題もありません。関数の外に見せたくないような一時的な変数はローカル変数として宣言すればよいのです。

これまで説明したのは一般的なグローバル変数とローカル変数の概念ですが，これらがどのように提供されているか（あるいは，どう使えばよいか）はプログラム言語によってまちまちです。以下ではC言語のグローバル変数とローカル変数について説明しましょう。C言語のグローバル変数とローカル変数をひと言で説明すれば，

グローバル変数：関数の外で宣言される変数
ローカル変数　：関数の中で宣言される変数

ということになります。たとえば，

```
int x;
main ( )
 {
   int a;
    :
 }
```

というプログラムがあるなら，main関数の外側で宣言されているint型変数xはグローバル変数，main関数の内側で宣言されているint型変数aはローカル変数です。

一般にローカル変数は関数の中でのみ有効な変数ですが，C言語のローカル変数はもっと局所的です。すなわち，C言語のローカル変数は関数の中ではなくブロックの中でのみ有効な変数です。ここでいうブロックとは｛と｝で囲まれた文の並び（複合文）のことです。ローカル変数はこのブロックの先頭で宣言されます（先頭以外では宣言できない）。そして，その変数が占めるメモリ領域は（概念的には）プログラムの実行の流れがブロックに入るときに生成され，ブロックを抜けるときに捨てられます[2]。

このように，ローカル変数はそれが宣言されたブロック内を実行しているときしか存在しない変数なのです。たとえば，

```
{int x; 文; 文; 文; ……}
```

というブロックでint型のxというローカル変数を宣言した場合を考えましょう。変数xの有効範囲はブロックの左端の｛から右端の｝までです。このxという変数はブロックの外側からは参照することができません。たとえば，

```
f ( ) {
    { int x; x=100; }
  printf("%d¥n",x);
}
```

という関数の定義があるとき，printf関数で値をプリントする変数xは，ブロック内で宣言され100という値を代入されている変数xではありません。この場合はprintf関数の引数になっている変数xはグローバル変数として宣言されていると考えます。

ところで，C言語の関数定義の本体はひとつの複合文（ブロック）といえます（形式はまったく同じですね）。このため，関数の先頭で宣言された変数は，その関数内だけで有効なローカル変数になります。関数の本体とは別の（関数の内部の）ブロックで宣言されるローカル変数が使用される頻度（非常に少ない）を考えれば，実質的にローカル変数を（ブロックではなく）関数の中で宣言される変数といっても不都合はありません。どうです，ちゃんとつじつまが合うようになっているでしょう。

さて，関数の本体を形成するブロック先頭で宣言するローカル変数はともかく，関数の本体とは別のブロックで宣言するローカル変数はどのように使用するのでしょう。このようなローカル変数は本当に一時的な変数が必要になるときに使用します。具体的には，

```
if(x>y)  { int tmp;
    tmp=x; x=y; y=tmp;
  }
```

などという例が考えられます。

この例は変数xの値が変数yの値より大きい場合（xもyもint型としておきましょう），xとyの値を入れ替えるという記述です。変数の値を入れ替えるためには片方の値を退避しておく一時的な変数がどうしても欲しくなります。

しかし，この変数は本当に一時的なものなので，関数の先頭でほかのローカル変数と同じレベルで宣言するのは気が引けます（もしかしたら，この部分で1度参照されるだけかもしれませんし）。ブロック内のローカル変数はこのような悩みを解決してくれるものなのです。そして，ブロック内でのローカル変数を上手に使えば，関数の先頭で行うローカル変数の宣言を本当に重要なものだけに絞り込むことができ，プログラムを読みやすくできるのです。

---

1) C言語を高級言語と呼ぶか否かは議論が分かれるところ。ただし，C言語の文法が高級言語的な性格を持っていることは確かである。

2) 関数の定義の中に別のブロックがあり，そのブロックの先頭でローカル変数が宣言されているとき，そのローカル変数の記憶領域がブロック内の文の実行が開始されるときに確保されるとは限らない。実際は，そのようなローカル変数の記憶領域も関数の先頭で宣言されたローカル変数の記憶領域と同時に確保されることが多い。これはプログラム実行時のスピードを上げるためであろう。

変数のスコープ（有効範囲）

ブロックの外部で宣言された変数とブロックの内部で宣言されたローカル変数が同じ名前のときもあります。このときはローカル変数のほうが有効になります（それが局所的という意味です）。もし，ブロックが入れ子になって，

```
{
  int x, y;
  x=2;
    {
      int x;
      x=1;
      y=10;
    }
    x=x+3;
}
```

というプログラム記述があったとすれば，内側のブロック（｛と｝の間）で宣言されたxというローカル変数は外側のブロックで宣言されているローカル変数のxとは無関係です。したがって，外側のブロックで変数xに2が代入され，内側のブロックで同じ名前の変数xに1が代入されたとしても，外側のブロックでの，

x+3

という式の計算では内側のブロックのx（=1）の値が使われることなく，外側のブロックでのx（=2）の値が使われるのです。

また，内側のブロックではyという変数に10という値を代入していますが，この変数は内側のブロックでは宣

言されていません。この場合はひとつ外側のブロックが参照されます。上の例では外側のブロックでローカル変数としてyが宣言されていますから，それが参照され，その変数yに10が代入されるのです。

もし外側のブロックでも変数yが宣言されていなかったらどうでしょう。そのときはさらに外側のブロックが探されます。こうして変数を探していったとき，どこにもその変数が宣言されていなければ，最後には関数の外部のグローバル変数にたどり着きます。そして，その変数が大域変数にも宣言されていない場合はエラーです。

ある変数を参照するとき，その変数がどこで宣言されているかをたどる順序を整理しておきましょう。これを変数のスコープ（有効範囲）といいます。ある変数を参照するとき，最初に検索されるのはその変数を使用しているブロック内です。次はそのブロックの外側にあるブロックです。こうしてブロックを外へ外へとたどっていくとそれは関数の本体を定義するブロックに突き当たります。そして，その次は関数の外部，といきたいところですが，その前にその関数の引数が検索されます。引数は（呼ばれる側の）関数にとって都合のいいように（呼ぶ側によって）初期化されたローカル変数とみなすこともできます[3]。そして，引数の次にやっと関数の外部のグローバル変数が検索されるのです。この関係の具体例を図1に示しておきましょう。

ところで，関数の定義を書く場合，うっかりして，

```
f(a,b)
int a,b;
{
   ⋮
}
```

と書くべきところを｛の位置を違えて，

```
f(a,b)
{
    int a,b;
    ⋮
}
```

としてしまうことがあります。どちらも文法的には間違いではありません（後者はint型の引数の宣言が省略されている形式と考えることができる）が，ここまで読んできた人には関数内で参照するａ，ｂという変数が前者と後者の場合でまったく別物であることがわかるでしょう。後者のように記述すると，関数内で参照する変数のａやｂは，引数ではなく，関数の先頭で宣言したローカル変数のａやｂのほうが参照されるので，引数の値を参照することができなくなってしまうのです。

3) 関数から呼び出し側に戻るときに引数の領域も捨てられる。その領域を呼び出し側が参照することはない。したがって，引数をローカル変数のように一時的な変数として使用することもできる（値を変更してもよい）。

自動的な変数と静的な変数

ローカル変数は原則的には自動的（automatic）です。つまり，関数に入った時点で領域が確保され，関数から出るときにその領域は捨てられてしまいます。一方，関数の外部で宣言されるグローバル変数は与えられた値を，それが変えられるまでいつまでも保持しています。このような変数を静的（static）であるといいます。

Ｃ言語ではすべての変数は（グローバルとローカルという分類のほかに）この自動的と静的の２つに分類することができます。この区別をＣ言語の文法では記憶クラスと呼んでいます。要するに，すべての変数はある瞬間にだけ存在する（自動的）か，永久に存在する（静的）かのどちらかなのです[4]。

さて，これらの変数の初期化について考えてみましょう。静的な変数はあらかじめ記憶領域が確保されていて，そこに初期値が格納されています。これはプログラム実行前（コンパイル時）での初期化です。それに対し，自動変数は宣言された時点で毎回記憶領域が確保されるので，そのときに初期化されます。これはプログラム実行中の初期化です。静的な変数の初期値はコンパイル時に行われますからその初期値は定数値（コンパイル時に計算可能な値）でなければなりません。しかし，自動的な変数については定数値によっても別の変数の値によっても初期化することができます（実行時に値が定まっていればよいので）。これが実行時に初期化を行うメリットです。

しかしながら，自動的変数の初期化は単に文の記述を省略するという意味しか持っていません。たとえば，

```
{int a=10；……}
```

という自動的変数の宣言時の初期化は，

```
{int a; a=10；……}
```

という表現の省略形にすぎません。実質的には単に簡潔さの違いだけなのです。

興味深いのは自動的な配列の初期化でしょう。

```
{int x [3]= {1,2,3}；……}
```

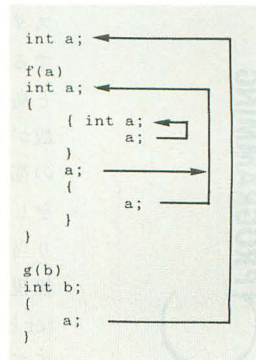
という記述は，

```
{int x [3]； x [0]=1； x [1]=2； x [2]=3；……}
```

と同じ意味ですが簡潔さは格段に違いますね[5]。

さて，先ほどローカル変数は自動的だといいましたが，ローカル変数を明示的に静的な変数として宣言することも可能です。そのときは，従来の変数宣言に対してstatic（静的な）というキーワードをつけます。たとえば，

```
ransu (  )
{
    static int r=1729；
```

図1　変数のスコープの例

```
        r=r * 131071;
        return(r);
    }
```

という疑似乱数を発生させる関数ransuの定義では変数rを静的なint型のローカル変数として宣言してあります。変数rは局所的変数でありながらstaticと宣言されたためにコンパイル時に領域が確保されます。そして，この領域は関数を抜けても消滅することなくいつまでも存在します。

いまstatic変数rには1729という初期値が与えられています。ransu関数は呼ばれるたびに変数rの内容を書き換え，その値を関数の戻り値としますが，その書き換えた値は次にransu関数が呼ばれたときに再び使用されます。このように変数rのひとつ前の値を使うということは，その変数が静的だからできる芸当です。もし，上のransu関数で変数rが自動的なら常に同じ値（1729＊131071）が戻り値として返ってきて乱数としての意味をなしません。まさに，静的変数さまさまです。

静的な変数を宣言するためには，staticというキーワードがありますが，逆に自動的であることを明示するための宣言もあります。それがauto（automaticの略）です。autoもstaticと同様に変数宣言の前につけて，

```
    auto int a;
```

という形式で使用します。しかし，グローバル変数を自動的と宣言することはできません（意味がない）し，ローカル変数も何もいわなければ自動的とみなされるので，autoというキーワードを使うことはまずないでしょう。

自動的変数は原則的にはスタック上に記憶領域が確保されます。しかし，C言語ではregisterというキーワードを変数の宣言につけることによって，自動的変数をCPU内部のレジスタに割り当てられるようになっています。このような変数をレジスタ変数といいます。CPUのレジスタは最高速で参照できるメモリみたいなものと思ってよいでしょう。このレジスタ変数の宣言はautoやstaticの宣言と同様です。たとえば，int型の自動的変数aをレジスタ変数としたい場合は，

```
    register int a;
```

と宣言します。

変数がレジスタにあればそれを参照するための時間はスタック（メモリ）よりも格段に速いので，頻繁に使用するfor文などでの制御変数はレジスタ変数に宣言すると高速な処理が期待できます。しかし，実際に自動的変数がレジスタへ割り付けられるかどうかはCコンパイラの都合で決まります。いくらプログラムでregister指定をしても適当なレジスタが空いていない場合は変数に割り当てることができません[6]。この場合はregister指定は無視されて，通常の自動的変数として扱われます。registerという指定はその変数を「なるべくレジスタに割り当ててください」という意味でしかないのです。もっとも，最近の賢いCコンパイラは自動変数をなるべくレジスタに割り当てようとするのでregisterという指定自体が意味をなさなくなってきているのも確かです。

レジスタ変数については，話題がいきなりハードウェアに近くなってしまいますし（スタックとかも出てくるし），最近のプログラミングではたいして意味のない概念なのでレジスタ変数を使いこなせるように努力する必要はありませんよ（一般教養程度にとどめておきましょう）[7]。

---

4) C言語でいう記憶クラスとは変数の存在期間と記憶方式を含む言葉である。自動的，静的とは存在期間のみしか問題にしていない。記憶クラスという言葉には変数をレジスタに割り付けるかメモリに割り付けるかということも含まれている。

5) GCCやANSIで許されているこの初期化の方法はXCではバージョン2で初めて許されるようになった。

6) たとえば8086用のCコンパイラでは，多くの場合，レジスタ変数用としてsi, diという2つのレジスタしか用意していないので3個以上のレジスタ変数は確保できない。

7) かつてはローカル変数のうちどれをレジスタ変数として宣言したらもっともプログラムの性能が出るかということがC言語プログラミングのテクニックに含まれていた。

関数の再帰呼び出し

前回，そして今回と，C言語の関数に関するさまざまな話題を取り上げてきています。これまでの説明で，

関数の定義のしかた

ローカル変数の使い方

は，だいたいわかったことと思います。そこで，今度は関数の一大トピックスである，

関数の再帰呼び出し

について学ぶことにしましょう。

大昔のプログラミング言語であるFORTRANやCOBOLでは不可能ですが，比較的最近のプログラミング言語ではどれも関数を再帰的に定義して使用する（再帰呼び出し）ことができます[8]。これは，関数が自分自身を直接あるいは間接的に呼び出す機能です。これはある処理（計算）をするのに，それよりも多少簡単な条件を自分に与えて実行した処理の結果を利用するというものです。

あまりによく用いられる例（カビが生えている？）で恐縮ですが，自然数の階乗計算は，与えられた自然数より1だけ小さい自然数の階乗計算に帰着することができます。すなわち，自然数nの階乗をfact(n)で表すと，

fact(n)はfact(n−1)から計算できる

というアイデアが再帰呼び出しです。わざわざいうまでもありませんが，fact(n)とfact(n−1)の関係は，

fact(n)=n＊fact(n−1)

となっています。100の階乗であるfact(100)を計算するためにはfact(99)を計算します。このfact(99)を計算するためには，同様にfact(98)を計算します。このように，

次々と1だけ小さい自然数の階乗を計算することにすれば，やがてはfact(0)を計算すればよいことがわかります[9]。このとき，fact(0)の値がわかっていれば，あとは将棋倒しの理論でfact(1)，fact(2)，fact(3)……，が計算でき，最後にfact(100)が計算できるのです。

階乗計算の例を見てわかるように，再帰とは問題をより簡単な条件，より簡単な条件での処理へと帰着し続けていく方法です[10]。ただ，条件をどんどん簡単にしていくだけでは終わりがありませんから，それをどこかで打ち切ってやる必要があります。これが停止条件と呼ばれるものです。階乗計算の例では，

fact(0)＝1

が停止条件として与えられていることが多いようです。関数で再帰呼び出しを行う場合の「条件」とは関数への「引数」ということになります。結局，関数を再帰呼び出しで定義する場合は，

1）引数がもっとも簡単な場合の処理を定義する。これが停止条件になる。

2）ある引数に対する処理を，もう少し値の小さい引数で自分自身を呼び出した処理で定義する。

ということを考えればよいのです。具体的には，再帰呼び出しをする関数の定義は，

```
func(n)
int n;
 {
    if(nが最小値) {
        自分を呼ばない単純な処理；
    }
    else {
        func(n−1)を使用した処理；
    }
 }
```

というイメージになるでしょうか。

ところで，C言語や多くのプログラミング言語で関数の再帰呼び出しができるようになっているのは，私たちの扱う問題の多くが再帰的な構造をしているからなのでしょう。つまり，問題をより簡単な条件に帰着させて解決できる場合が多いからだと思われます。特に，

・ハノイの塔
・巡回騎士の問題
・エイトクイーン
・宣教師と人食い土人

といった有名なパズルの解法には再帰呼び出しを使ったものが圧倒的に多いようです。

また，C言語はOSやコンパイラなどの言語プロセッサを記述するためのシステム記述言語として誕生しました。プログラミング言語の構文解析にも再帰呼び出しは不可欠です。プログラミング言語の文法でよく見掛ける式の構文は，

式：＝項　　または　式＋項
項：＝要素　または　項＊要素
要素：＝変数名　または
　　　　数値　　または
　　　　（式）

といったものです[11]。

こういった構文を解析する場合は，「式」を解析するための関数，「項」を解析するための関数，「要素」を解析するための関数を用意してそれぞれの解析を分担させます。見てわかるように，ここでの構文では，「式」は「式」から，「項」は「項」から再帰的に定義されています。さらに「要素」はそれを定義する大もとであったはずの「式」から定義されています(間接的な再帰)。このとき構文解析を行う関数を定義しようとしたら，再帰呼び出しのオンパレードになることが予想されます。もし，関数の再帰呼び出しができないとしたら，このような構文の解析はかなり困難でしょう。

8）UNIX上のFORTRANであるf77ではサブルーチンや関数の再帰呼び出しができる。時代は変わった。

9）最近の自然数論では0も自然数とみなすのが普通です。

10）再帰は帰納とも呼ばれる。どちらもrecursionの日本語訳。数学で出てくる数学的帰納法（英語ではmathematical induction）も本質的には再帰呼び出しと深い関係がある。帰納という言葉の語源としては次の説が有力（ウソ）。明日を知るためには今日を知る必要がある。今日を知るためには昨日を知る必要がある。昨日，昨日へとさかのぼっていくから帰納（昨日）法。

11）この構文をそのまま解析する関数を作ると無限ループに陥る（停止条件に行き当たらない）はず。それを避けるために，

式：＝項　　または　項＋項＋項＋…
項：＝要素　または　要素＊要素＊…
要素：＝変数名　または
　　　　数値　　または
　　　　（式）

などという構文に変換して解析を行う。こうしても，「要素」がその大もとの「式」から（間接的な）再帰的に定義されていることに変わりはない。

◆基礎力を高めよう

設問　次のプログラムにおいて，それぞれのprintf関数がプリントする値を答えてください。

```
int x;
main ( )
 {
        x=1；
        while(x<2) {
            int x=1；
            f ( )；
          printf("%d¥n",x);       …(1)
        }
        printf("%d¥n",x);         …(2)
 }
f ( )
 {
        x=x+2；
```

```
        printf("%d¥n",x);          …(3)
        g(x);
}
g(x)
int x;
{
        int x=x;
      printf("%d¥n",x);            …(4)
}
```

（解答は80ページに示します）

再帰呼び出しを行うプログラム

それでは，今回学んだ項目の復習として再帰呼び出しを行うプログラムを作ってみましょう。単純な再帰のプログラムと複雑な再帰のプログラムの例として，

・第6回の連載で作成した数値を文字列に変換するプログラムの再帰版（単純な再帰）

・与えられた構文を持つ式を解析して値を求める電卓プログラム（複雑な再帰）

を考えます。それぞれを以下に説明します。

●数値を文字列に変換するプログラム

数値が与えられたとき，それを文字列に変換して特定のchar型配列に格納するプログラムを作ります。たとえば，123という数値が与えられると"123"という文字列を特定の配列に格納するプログラムです。

これはC言語のライブラリ関数であるatoi関数の逆操作になっています。これと同じ働きをするプログラムはこの連載の第6回（1991年4月号）で作りました。そのときは，与えられた数値をどんどん10で割っていったときの余りを文字に変換しながら配列に押し込み，最後に配列の要素を逆順に並び替えるというアルゴリズムを使用しました。このときは，プログラムのアルゴリズムが単純である半面，一度求めた配列要素を逆転するという処理が入ってくるので美しさに欠けていました。一方，再帰呼び出しを用いてこのプログラムを書けばずっとエレガントなものになります。

引数として与えられた数値を文字列に変換して特定の配列に格納する関数をmakestrという名前にしましょう。この関数があるとすれば，makestr(123)という処理は，与えられた数値を10で割った商を引数にして再帰呼び出しを行う，

makestr(12)

を実行したあとに，与えられた数値を10で割った余りを文字に変換（'0'を加算する）した'3'を配列の最後に書き込む処理に帰着できます。makestr(12)を実行したあとは配列内には"12"という文字列が格納されているはずですから，この方法でよいことがわかりますね[12]。

makestr(12)という処理は同様に，

makestr(1)

を実行したあとに，'2'という文字を配列の最後に書き込む処理に帰着できます。次はmakestr(1)という処理を行うことになりますが，引数は1桁の数値ですから，これは簡単に'1'という文字に変換できます。このときは配列の先頭に"1"という文字列を書き込むだけでおしまいです。これが再帰呼び出しの終了条件になります。

もし，一番最初に与えられた数値が負の数であれば，その絶対値をmakestr関数の引数として渡すことになります。このとき，makestr関数を再帰呼び出ししていって終了条件に達したならば，まず'－'という文字を配列に書き込んでから引数を変換した文字を書き込むことになります。

ところで，文字を配列に書き込むときには書き込む位置（配列の添字）を保持する変数が必要になります。この添字はグローバル変数として定義するとよいでしょう。配列への書き込みが行われるたびにグローバル変数の添字を1だけ更新することにしておけば，書き込みが行われるたびにその順で文字が配列に格納されていきます。また，もとの数値の符号はmakestr関数への引数とすることもできますが，グローバル変数として宣言しておくのが簡潔でいいでしょう。

このような方針で書かれたプログラムがリスト1です。連載の第6回で示したプログラムよりもエレガントになっているのがわかるでしょうか。

---

12) ここで定義しようとしているmakestrという関数で作られる文字列は，char型配列の中に文字が並んでいるだけで，文字列の最後を示すコード（0）は入っていない。

●電卓プログラム

ここで考える電卓プログラムが認識することのできる式の構文は，先に示した例を少し変更した，

式　:＝　項　または　項±項±項±…
項　:＝　要素　または　要素＊/要素＊/要素＊/…
要素：＝　数値　または　(式)

注）＊/は＊か/を示すものとする

であると仮定します。これは「式」の構成要素である「要素」がもとの「式」によって定義されている間接的な再帰の例です。この構文で定義される式は数値と四則を示す演算子（+，－，＊，/）およびカッコからなる，

1+2＊3－(1+2)＊(3+4)

といった，いわゆる普通の四則演算の計算式です。この例を上の式の構文に当てはめると，

1，2＊3，(1+2)＊(3+4)

のそれぞれが「項」となり，

1，2，3，(1+2)，(3+4)

が「要素」となることがわかりますね。そして，この「要素」のうち，カッコで囲まれた，

1+2，3+4

は，さらに「式」の構文をしていなければなりません。

さて，このような構文解析を行うプログラムでまず必要なものは，入力された1行からシンボルや記号（これらをトークンという）をひとつずつ切り出してくるトークンリーダ（シンボルリーダともいう）と呼ばれる関数です。トークンリーダは読み込んだシンボルや記号の情報を，通常は，グローバル変数に格納したり，自らの戻り値として返してきます。構文解析を行う関数はこのトークンリーダから返される情報を元にして構文の解析を行っていきます。

そこでまずトークンリーダを書かなければなりませんが，ここでは次にくるトークンが「数値であるか記号であるか」という情報とそのトークンの「値（数値あるいは記号)」を返してくれれば十分です。いまは，トークンリーダをnext_tokenという関数として，この関数を呼ぶたびに，トークンが「数値であるか記号であるか」の情報がtoken_typeというグローバル変数に，トークンの具体的な値がtokenというグローバル変数に格納されるものとしましょう。たとえば，

```
    1+2*3
```

という式（文字列として与えられる）が入力されているときは，nexttoken関数を呼ぶたびに，

```
token_type＝数値    token＝1
token_type＝記号    token＝'+'
token_type＝数値    token＝2
token_type＝記号    token＝'*'
token_type＝数値    token＝3
```

という情報が順番にそれぞれのグローバル変数に格納されるものとします。すでに文字や文字列の操作方法を学んでいる私たちにとって，このようなトークンリーダを書くことはそれほど難しくないでしょう。

トークンリーダができたら，次は構文解析を行う関数を書きます。「式」，「項」，「要素」を解析する関数をそれぞれ，expr, term, elementとしましょう。それぞれの関数の戻り値は「式」や「項」や「要素」を解析して求められる数値であるとします。このとき，求める電卓プログラムは計算式を1行入力してからexpr関数を呼び出し，その関数から返される値をそのままプリントするという非常に簡単なものになってしまいます。また，expr, term, elementという関数自体も式の構文をそのまま書き下したように簡単に定義することができます。すなわち，

```
expr（ ）    /*  「式」の解析  */
{
    int  x, y；
    term（ ）を呼びxに代入；
    while(次のトークンが'+'か'−'){
        term（ ）を呼びyに代入；
        if(トークンが'+')
            x＝x＋y；
        else
            x＝x−y；
    }
    xを戻り値とする；
}
term（ ）    /*  「項」の解析  */
{
    int  x, y；
    element（ ）を呼びxに代入；
    while(次のトークンが'*'か'/') {
        element( )を呼びyに代入；
        if(トークンが'*')
            x＝x＊y；
        else
            x＝x/y；
    }
    xを戻り値とする；
}
element（ ）    /*  「要素」の解析  */
{
    int  x, y；
    if(次のトークンが数値) {
```

リスト1 数値→文字列変換プログラム

```
 1: /*
 2:           整数→文字列  変換プログラム（再掲版）
 3: */
 4: int     sign;                 /* 符号を保持するグローバル変数 */
 5: char    string[100];            /* 特定のchar型配列        */
 6: char    index;                /* 添字を保持するグローバル変数 */
 7:
 8: main()
 9: {
10:     int     number; /*  入力される数値を保持するローカル変数 */
11:
12:     scanf("%d",&number);      /* 整数の読み込み */
13:     printf("整数=%d ",number);  /* 読み込んだ値をプリント */
14:
15:     sign=0;                   /* 符号を初期化      */
16:     if(number<0){
17:         sign=-1;              /* 負数であることを記憶 */
18:         number=-number;         /* 正数に変換          */
19:     }
20:     index=0;                  /* 添字を初期化      */
21:     makestr(number);            /* 変換処理       */
22:     string[index]=0;            /* 終了コードを書く */
23:
24:     printf("文字列=%s¥n",string);/* 変換した文字列をプリント */
25: }
26:
27: makestr(n)
28: int n;
29: {
30:         int q,r;
31:         if(n>=0 && n<=9){                  /* 引数が1桁ならおしまい */
32:             if(sign<0){            /* 負数だったら符号を書く */
33:                 string[index]='-';
34:                 index=index+1;
35:             }
36:             string[index]=n+'0';     /* 引数を文字に変換する */
37:             index=index+1;
38:         }
39:         else{
40:             q=n/10;               /* 10 で割った商    */
41:             r=n%10;               /* 10 で割った余り */
42:             makestr(q);           /* 商を変換する     */
43:             string[index]='0'+r; /* 余りを文字に変換 */
44:             index=index+1;
45:         }
46: }
```

実行結果

```
整数=0  文字列=0
整数=1  文字列=1
整数=-1  文字列=-1
整数=-123  文字列=-123
整数=23345  文字列=23345
```

注）キーボードから入力した内容は見えていない

```
                数値を戻り値とする；
            }
            else {
                if(次のトークンが'(')
                    expr（　）の値を戻り値とする；
                else
                    エラー；
            }
        }
```

がそれぞれの関数の概念的な定義です。ただし，実際にはこれらの定義の中にエラー処理などが入りますからもう少し複雑な定義になります。

以上の方針で書かれたプログラムがリスト2です。計算式の構文解析という，いかにも難しそうなテーマのプログラムですが，現実はこの程度のものに過ぎません。

＊

今回はローカル変数とグローバル変数の使い分け，および再帰呼び出しを行う関数の定義の仕方について説明してきました。ここまでくればC言語の解説も一段落です。たいていのプログラムならばこれまでの知識で書けるはずです。これから先は，より高い応用へ進んで行ってもよいのですが，その前にこれまで学んだことをじっくり復習して完全に身につけてしまいましょう。そこで，次回から数回は，復習の意味も込めて，これまでの知識の補足という観点で解説してみたいと思っています。とりあえず，次回は式と演算子の使い方を取り上げる予定です。それでは，来月までさようなら。

◆基礎力を高めようの解答

設問

1) 1，　2) 3，　3) 3，　4) 不定

解説

変数のスコープを理解していれば難しくはない。ただ4)の不定というのは意地が悪い出題だったかもしれない。ただ，4)については興味深い事項が含まれている。4)のprintfで参照される変数は引数ではなく，関数の本体で宣言されるローカル変数であることはすぐわかる。ただし，その変数を初期化する，

int x=x;

という宣言が曲者である。=の右側のxが関数の引数であるxと思い，引数の値が宣言したローカル変数にコピーされると考えた人は考えすぎ。この宣言が，

int x; x=x;

の省略形であることに気づけば，どちらも同じローカル変数を示していることがわかる（自分自身への代入）。ローカル変数（自動的変数）の初期値は不定である。

ところでXCでは引数と同じ名前のローカル変数を宣言すると変数の2重定義でエラーになる（コンパイルできない）。GCCでは引数の値が見えなくなりますという警告になる（コンパイルできる）。XCの挙動は明らかにおかしい。

リスト2
電卓プログラム

```
  1: /*
  2:             電卓プログラム（あるいは数式の構文解析）
  3: */
  4: char line[100];
  5: int  index;
  6: int  token_type;
  7: int  token;
  8:
  9: main()        /* 本体：リターンのみの入力で終了 */
 10: {
 11:     while(1){
 12:         printf("? ");      /* line に1行読み込む */
 13:         if(gets(line)==0) break; /* EOF */
 14:         if(line[0]==0)    break; /* リターンのみ */
 15:         index=0;
 16:         next_token();
 17:         printf("%s=%d¥n",line, expr());
 18:     }
 19: }
 20:
 21: next_char()      /* line というchar型配列から1文字を読む */
 22: {
 23:     int ch;
 24:     ch=line[index];
 25:     index=index+1;
 26:     return ch;
 27: }
 28:
 29: next_token()      /* トークンリーダ */
 30: {
 31:     int i,n,ch;
 32:     while(index<100){         /* まずは空白でない文字を探す */
 33:         ch=next_char();
 34:         if(ch==' ')  continue;  /* 空白をスキップ */
 35:         if(ch=='¥t') continue;  /* タブをスキップ */
 36:         break;
 37:     }
 38:     if(ch>='0' && ch<='9'){ /* 最初の文字が数字なら，数値を得る */
 39:         n=ch-'0';
 40:         while(1){
 41:             ch=next_char();
 42:             if(ch<'0' || ch>'9') break;
 43:             n=n*10+(ch-'0');
 44:         }
 45:         index=index-1;  /* 読みすぎた分を戻す */
 46:         token_type=0;   /* 数値なら0をセット */
 47:         token     =n;   /* 数値を退避        */
 48:     }
 49:     else{
 50:         token_type=1;   /* 数値以外（記号）なら1をセット */
 51:         token=ch;       /* 記号を退避 */
 52:     }
 53: }
 54:
 55: expr()  /* + と - の処理 */
 56: {
 57:     int     p,op,q;
 58:     p=term();
 59:     while(token=='+' || token=='-'){
 60:         op=token;
 61:         next_token();
 62:         q=term();
 63:         if(op=='+') p=p+q;
 64:         else    p=p-q;
 65:     }
 66:     return(p);
 67: }
 68:
 69: term() /* * と / の 処理 */
 70: {
 71:     int     p,op,q;
 72:     p=element();
 73:     while(token=='*' || token=='/'){
 74:         op=token;
 75:         next_token();
 76:         q=element();
 77:         if(op=='*') p=p*q;
 78:         else{
 79:             if(q==0){
 80:                 printf("ゼロ除算¥n");
 81:                 exit(1);
 82:             }
 83:             p=p/q;
 84:         }
 85:     }
 86:     return(p);
 87: }
 88:
 89: element() /* 数値とカッコの処理 */
 90: {
 91:     int     p;
 92:     switch(token_type){
 93:     case 0: /* 数値 */
 94:         p=token;
 95:         next_token();
 96:         break;
 97:     case 1: /* 記号 */
 98:         if(token=='('){
 99:             next_token();
100:             p=expr();       /* 再帰！！！！ */
101:             if(token!=')'){
102:                 printf("式に)がない¥n");
103:                 exit(1);
104:             }
105:             next_token();
106:         }
107:         else{
108:             printf("式が誤っている¥n");
109:             exit(1);
110:         }
111:         break;
112:     }
113:     return(p);
114: }
```

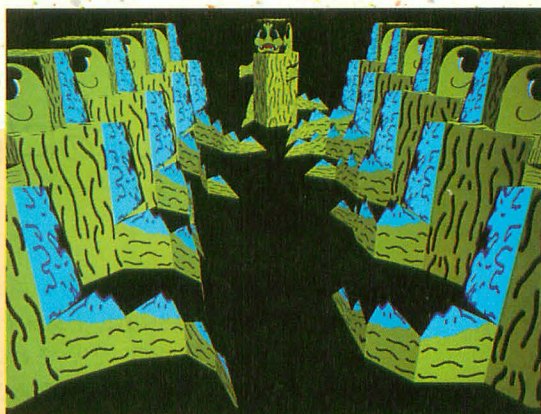
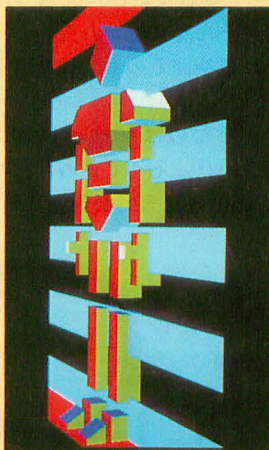
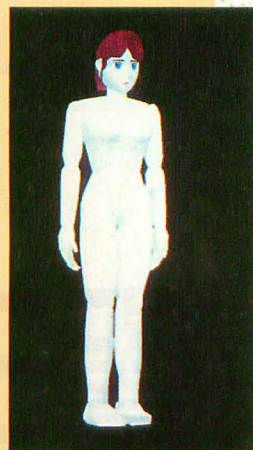
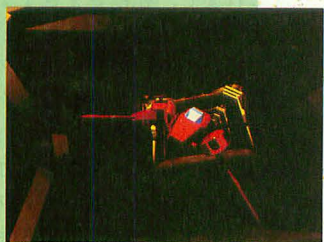
実行結果

```
?
1+2=3
?
1+2*3=7
?
(1+2)*(1+3*(4+5))=84
?
```

注）キーボードから入力した内容は見えていない

DōGA・CGアニメーション講座

今月から2回にわたって人体モデルのお話です。多関節構造体のノウハウをしっかりと学びとってください。
まずは，1990年9月号でもお馴染みのハコジラ君。今月号では，このハコジラ君のシッポを振らせるようにします。
そして，来月にはいよいよ人体モデルを動かすわけです。標準人体モデルとその例を紹介しておきますね。
さて，と。先日盛大に行われたアマチュアCGAコンテスト。しかし，記事中でも触れたとおり，オープニングのアニメーションは日の目を見ることができなくなってしまいました。それではあまりにかわいそう，というわけで，この場を借りてほんの少しだけ紹介しちゃいます。ほかでは見ることのできない作品ですよ。
最後に，関数の応用の応用として，きれいな薄桃色の桜をお届けしましょう。これは162ページを参照してくださいね。

チェックのタイルの上を走っている人体モデル。関節の動きをよく研究してください

お馴染みハコジラのシッポがうねうねうね……

これが標準人体モデルです。背景に幅10のストライプを入れてみました。身長がちょうど100になってますね。また，正面を赤，左側面を緑，上面を青にすることで，ねじれても軸がよくわかります

今月はまだ立っているだけです。来月号で頑張ってもらいましょう

これがまぼろしの（？）オープニングアニメです。配布版のビデオにも入っていません

季節感をまったく無視したサンプルです。単純な部品を組み合わせて，桜の花を作ってみました

［第1回］
3次元CGってなあに？

何年前のことだったでしょうか。はじめて私が3次元CGの画像を見たのは，ナショナル(パナソニック)のコマーシャルでした[1]。それは，ワイヤーフレームの紙飛行機がオフィスをあとに，シカゴの高層建築の中を飛び回るというものでした。CMがオンエアされるたびに見入ったのを憶えています。なぜそんなに魅かれたのか，……紙飛行機の飛んでいる空間そのものに魅力を感じたからでした。

CMを制作したのはロバート・アンド・エイブル・アソシエイツ[2]。そこで使われたコンピュータはVAX/11-750で，当時数億円はしたはずです。3次元CG画像の作品をつくりたいと思っても個人には夢のまた夢，そんな時代でした。

時は流れました。ここに掲載してある画像は，レイトレーシング[3]のソフトウェアを用いてX68000でつくったものです。少なくとも静止画では大型コンピュータを使った場合に匹敵する作品ができるようになりました。

3次元CGの作品をつくるのはとても楽しいことです。ちょうど自分でつくった無限の箱庭で遊ぶといった感じでしょうか。この楽しさを「響子 in CG わ～るど」を通じて，少しでもみなさんに知っていただければうれしく思います。

響子inCGわ～るど

3次元CGで作品をつくるということ

スケッチブックに絵筆で花を描く。粘土で動物をつくる。プラモデルを組み立てる。これらの作品をつくる行為に共通するのは，「手などの肉体を感覚的に覚えて使う」ことです。スケッチブックに花を描く場合を考えてみましょう。まず描いてみる。うまく描けなかった。もう一度描いてみる。こうしたことを何回も繰り返して，自分なりに納得できる作品が完成します。

ペイント系ソフトやドロー系ソフトなどの2次元CGについても同じことがいえます。マウスを使ってCRTに描く行為は，絵筆を使ってスケッチブックに描く，あるいは自在定規を使って描くのによく似ています。

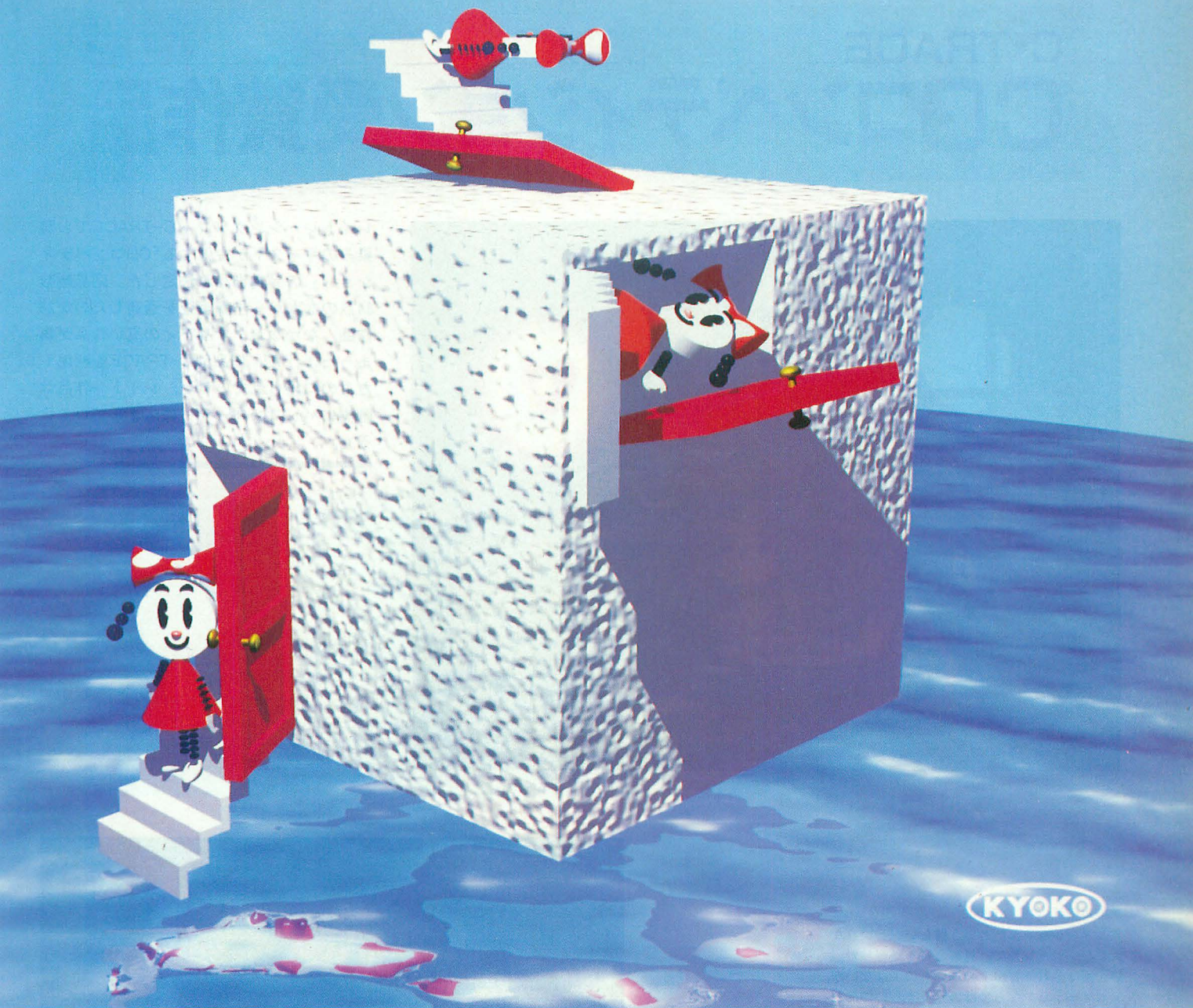
3次元CGではどうでしょうか？ X68000で使えるソフトウェアでは，「手などの肉体を感覚的に覚えて使う」といった行為は存在しません。3次元CG画像を構成する形や色，光源などの要素はすべて数字などのデータによって入力されます。入力には主にキーボードを使いますが，マウスでもできます[4]。

作品をつくる過程をかんたんにお話しましょう。「四角いオブジェをひとつ3次元空間に浮かばせたい」と脳でイメージするとします。次に立方体のX, Y, Z軸方向の大きさと位置さらに質感など[5]を決めて，データを入力します。あとはコンピュータが画像を描きだすのを待ちます。描かれた画像が気に入らなければデータを入力し直します。何回か繰り返して，自分なりに納得できる画像が完成します。[6]

このとき，(空間に立体が浮かんでいるイメージ)＝(数字などによって与えられた立方体や空間のデータ)＝(描かれたCG画像)の関係が成立しています。3次元CGで作品をつくるということは，「脳のなかのイメージを数学的なデータに置き換えて，そのデータを入力する」ことといえるでしょう。

これまで，私はさまざまな画材を仕事で使ってきました。色鉛筆でささっと描いたり，絵筆で緻密に描いたり，カラーインクをエアブラシにつめて描いたりしてきました。そのどれもが，完成するまで作品につきっきりでいなくてはいけません。3次元CGではデータさえ入力してしまえば，あとはコンピュータが描いてくれます。ひとことも文句を言わずに……。「2001年宇宙の旅」のHALに対するチャンドラー博士の気持ちがよくわかるような気がします。

プロフィール

寺尾響子(てらおきょうこ)
イラストレーター。X68000 EXPERT-HDを利用してCG制作に取り組む。DōGA主催のアマチュアCGAコンテストでも入選されたのは記憶に新しい。

たしかに　その場所は　ある

どこに？

ここに

わたしの中に

そこには　形と色　光と影　空気と風　未来と過去　そして空間がある

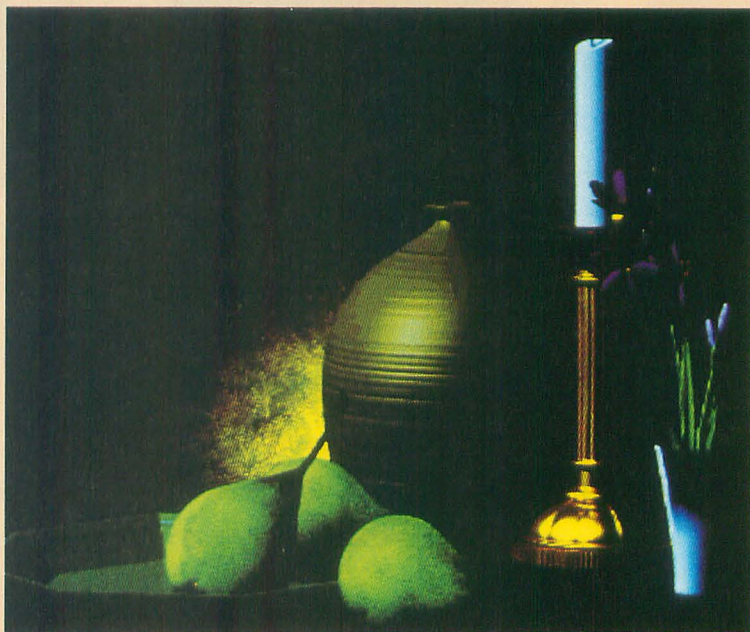
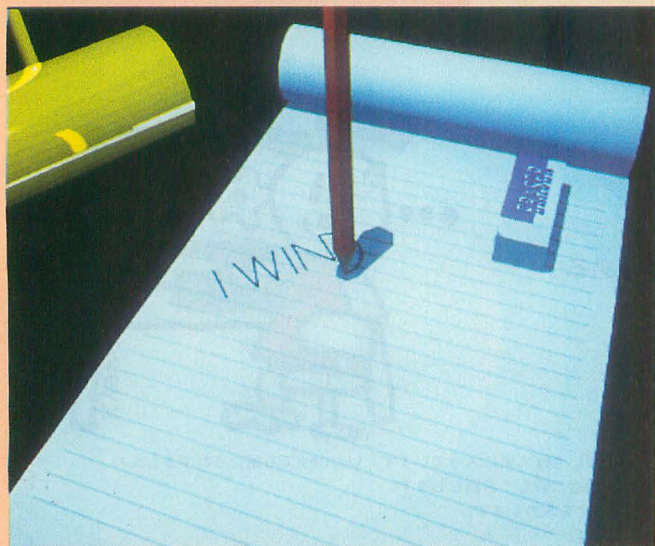
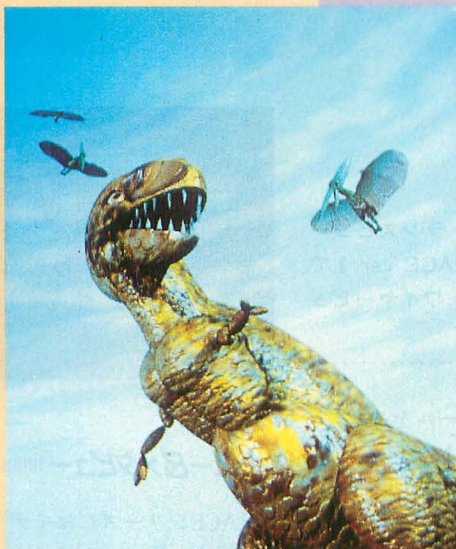
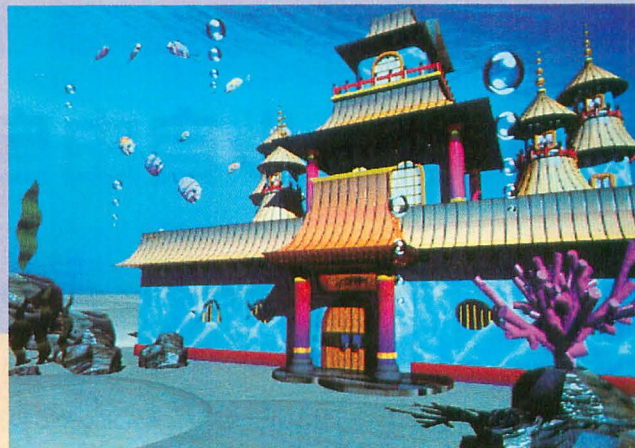
無限に続く空間がある

1)　そのころ，テレビゲームの元祖ともいうべき「スペース・インベーダー・ゲーム」が流行していました。
2)　ディズニーの「TRON」のCGもロバート・アンド・エイブル・アソシエイツの制作したものです。
3)　ターナー・ウィテッドが考案した光線追跡のアルゴリズム。計算時間がかかるけれど，画像が美しいのでいちばん気に入っています。以前Oh!Xでも紹介されてましたね。
4)　CADでポリゴンデータを入力するときは，マウスを主に使います。CADやポリゴンについてはまた別の機会に詳しくお話しします。
5)　実際には，この他にマッピングデータ，視点，光源，形状定義のための論理演算式などさまざまなデータを入力します。
6)　写真は1280×1024ドットのフルカラー（1600万色）画像を5×4インチのポジフィルムに出力したものです。

C-TRACE CGコンペティション受賞作品

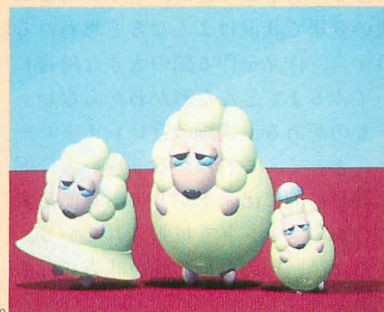
レイトレーシングソフトC-TRACEでお馴染みのキャストが主催する「CGコンペティション」の受賞作品が決定した。応募総数84点のなかから予備審査を通過したのが26作品，なかなかクオリティの高い作品が集まったようだ。作品はC-TRACEを利用したものに限られているが，レイトレ作品は確実にパソコンユーザーに広まっているようだ。

◀牛澤敏彦（38）X68000
エミリンの誕生日にブタ君とタヌキ君がケーキとプレゼントを持ってやってきた。愛娘に対する思いが込められた光と影のファンタジー。技術的にも優れているが，見る者に訴えかけるイメージが素晴らしい。

▼小川拓延（26）X68000
アニメーション部門という難しい（というか大変な）ジャンルで，見事金賞を獲得した。静止画はそれほど密度の高いものではないが，アニメーションならではの見て楽しい作品だ。

▲稲見薫（34）PC-9801RX21
メタボールのテストをかねて，ということだが，実に完成度が高い絵だ。油絵風の色彩と静的な構図が見事に調和している。どこの客間の額縁に飾ってもおかしくない。

▶神谷淑貴(28)
PC-9801RA21
あざやかな色彩感覚が龍宮の幻想性を強調している。水中の表現も見事な作品。

◀荒井清 (31) X68000ACE-HD
モデリングにもマッピングにもこだわりまくった秀作だ。

▲林秀則 (20) PC-9801VM2
計算時間だけで50日間もかかった涙の超大作。

▲畠山尚 (16) X68000ACE
プリミティブの組み合わせに力を注いだ作品。

▲桐谷佳典 (25) X68000 湾岸に巣くう怪物

▶矢野良 (35)
X68000SUPER
とぼけた羊君のアニメーションが楽しい。

▶熊野務 PC-9801RA2
動物たちのキャラクターデザインが個性的。

▼下田達也 (24)
X68000EXPERT II
くもりガラスの向こう側が作品のポイント。

▼河本保 X68000
1匹のカエルが実に強力なイメージだ。

CGコンペティション受賞者

応募作品総数 84点／審査対象作品数 26点

賞	受賞者	作品名
グランプリ	牛澤敏彦	Happy Birthday
準グランプリ	稲見薫	洋梨のある静物
金賞	小川拓延	Pencil
銀賞	神谷淑貴	Welcome to Ryugu
銅賞	荒井清	DINOSAURUS
ステゴ賞	林秀則	アニバーサリー
ステゴ賞	畠山尚	パイプオルガンよ永遠なれ
ステゴ賞	桐谷佳典	オイルマスター

●協賛会社特別賞

賞	受賞者	作品名
シャープ賞	矢野良	E・T・O
ソニーコンピュータシステム賞	熊野務	遠方見聞録 2
アイ・オー・データ機器賞	下田達也	くもり
月刊アスキー賞	河本保	カエル

審査員
長谷川一光　CGキッチンまざあぐうす代表
塩沢佐千子　C-TRACEユーザークラブ会長　東京芸術工学院講師
玉手峰人　超能力者
井川英雄　イラストレーター
協賛会社代表／キャスト　スタッフ

C-TRACE TP+

メタボール自由自在

Tan Akihiko
丹 明彦

C-TRACEシリーズの最上位バージョン。トランスピュータでメタボールが扱える。これまでのC-TRACE ver.3で拡張された機能はすべて含み，メタビュー，ワイヤービューといったプレビュアも加えられている。

1991年3月号で紹介したC-TRACE＋がようやくトランスピュータに対応した。

C-TRACEはレイトレーシングソフトウェアである。

C-TRACEの上位バージョンでメタボールを扱えるようにしたのがC-TRACE＋。それと並行してC-TRACEにはトランスピュータ対応バージョンというのもあり，こちらは計算専用のトランスピュータボードを使っているので画像を計算するのがとにかく速い。そして今回，C-TRACE＋がトランスピュータで動くようになった。メタボールはかなり重い処理で，本体だけで描画するのは少々苦しかったが，トランスピュータ版の登場で状況はよくなると思われる。

それから，作品を作る際の大きな障害に「描いてみるまでどんな形かわからない」というものがある。描画の遅いレイトレーシングにとって，これはかなり痛い。今回は，本計算の前に出来上がりの形を簡易表示するためのコマンドも拡張されている。これまた作品制作のための強い味方になることであろう。

メタボール

メタボールは変形・融合をその最大の特徴とするプリミティブ。通常のレイトレにおける造形が単純な図形を論理演算で組み立てて目的の形にしているのとは異なる。正の重みや負の重みを持つメタボールを適切に配置し，変形のぐあいをコントロールすることで目的の形を作り上げる。

X68000本体のみではどうしようもなく遅いメタボール。数値演算プロセッサボードを載せても快適とはいいがたい。トランスピュータ対応バージョンは当然待たれていたところであった。

トランスピュータで計算させてみて，改めてメタボールの処理がとんでもなく重いのだということを認識した。メタボールの処理の重さを知識として持っている人，そしてなによりもX68000本体だけでメタボールを扱った経験のある人ならば，決して遅いとは思わないだろう。しかし，僕には初めてトランスピュータが導入されたときに覚えた感動があるので，もしかするとメタボールといえども一瞬のうちに描画してしまうのではないかと期待してしまった。さすがにそれは甘かったようだ。

それでもかなり快適な速さであることに変わりはない。試し計算をするのに数分かかるのと数時間かかるのとでは，作業の能率もかなり違う。数分というのはまだ待てる時間だからだ。

メタボールを制御できるようになるには，経験を積むしかないように思われる。適当に投げ込むだけでは，面白い形は山ほどできるのだが，狙った形に作るのはかなり難しい。必然的に試行錯誤の作業となる。描画速度次第で作品の質まで変わる可能性もあるのである。

ワイヤビュー&メタビュー

C-TRACEシリーズにはモデラーがない。初めの頃はあったのだが途中のバージョンから姿を消している。したがって形状データはテキストエディタを用いて記述する必要がある。このおかげで，C-TRACEでは描画してみるまで形が確認できない。長いこと計算させてできてきた絵がちょっとした座標のずれのせいで台なしになっていたときなど，心底モデラーがほしいと思う。また，今後メタボールを導入していくことになるのだが，メタボールは制御が難しく，紙の上で設計したとおりに変形してくれるとは限らない。

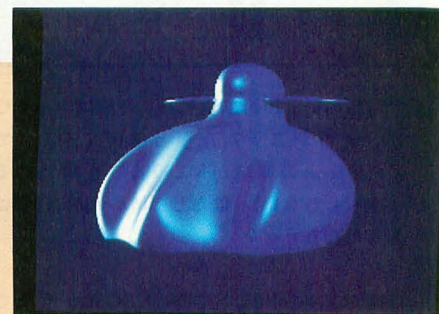
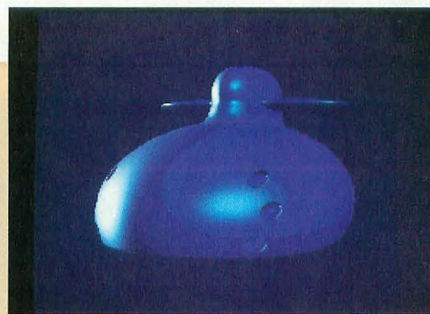
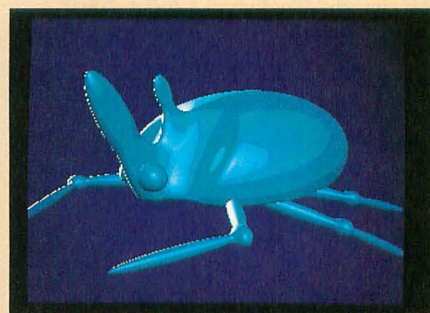
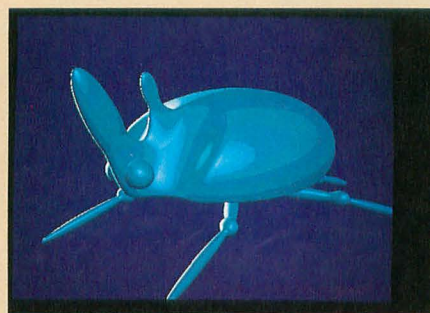
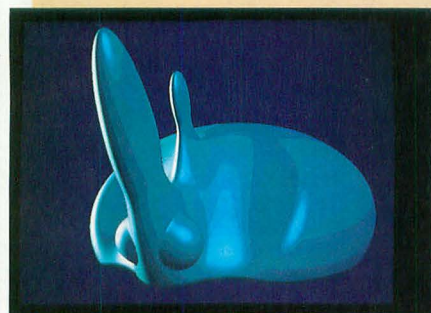
これを少しでも解決すべく，本計算に先立って大まかな表示をさせるためのコマンドが用意された。いわゆるワイヤビュー，およびメタビューである。

単なる表示コマンドだからモデラーのように対話的ではない（逆にいえば表示させたいときしかさせないですむ）。モデリング作業は相変わらずテキストエディタとC-TRACE＋のあいだを行ったりきたりすることになる。それでもこのようなプレビューができると助かる。

レイトレの試し計算は，時間のかからないように解像度を粗くしてするのがふつうなので，細かいところまで見ようと思うと試し計算にもけっこう時間を食う（ただし，

X68000用（ボード付き） 398,000円（税別）
キャスト ☎03（3705）1065

この領域だけを……

細かく計算することができる

メタボールの作例その1。潜水艦かな？

メタボールの作例その2。カブトムシだ。計算時間は3，4時間

C−TRACE＋にはスコープ機能も用意されていて，局所的に解像度を細かくして計算することも可能)。ワイヤビューやメタビューは，解像度の点については心配いらない。ワイヤビューはお決まりの緯線経線のワイヤフレーム表示。メタビューはちょっと変わっていて，適当な間隔で切った断面を表示し，全体の形状を把握させるようになっている。奥の方の断面は暗く，手前は明るく表示されるので，奥行きもつかみやすい。

速度の点ではどうかというと，プレビュー機能として使うならもう少し速いほうがいいかなというところ。メタボールはその性質上，単体ではメタボールたりえない。数が多くなると相互干渉も激しくなる。一瞬で終わるような簡単な計算ではないというのはわかるが，それでももっと速くなってくれたほうが使いやすい。数分間かかるので，解像度が粗くて用が足りるうちは試し計算のほうが速いこともある。

理想をいうならば，リアルタイムで変形してくれるようなモデラーが最高。あちらこちらを引っ張って，まさに粘土を扱う感覚で作業ができる（性質的にはメタボールと粘土はまったく異なるものだが)。そこまでやれば，メタボールをよく理解していない人でも確実に形状を制御できるようになるだろう。

なお，ワイヤビューやメタビューは今回新しく追加された部分だが，ノーマルのC−TRACE＋にも装備されることになる。

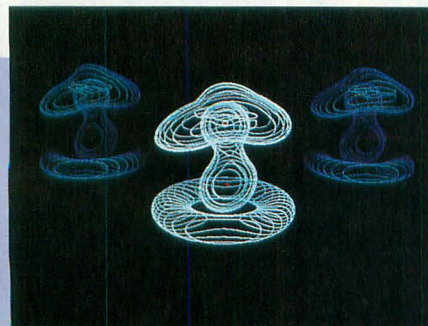
最後に

トランスピュータ版は速くなるというだけで，通常のC−TRACE＋とまったく同じである。詳細は3月号のC−TRACE＋のレビューをご覧いただくことにしよう。

トランスピュータを使っていると思うことがある。トランスピュータを使うレイトレは，X68000を占有する時間はとても長いが，そのあいだX68000自身はほとんど働いていないのではないかと。計算はトランスピュータが一生懸命にやっている。X68000はその結果をもらってきて表示するくらいのことしかしない。それだけのために何時間も，あるいは何日ものあいだ，X68000をほかのことに使うことができない。

そこでまた思った。SX−WINDOWで動かしてはどうかと。トランスピュータを使っていれば，疑似マルチタスクもそれほど足かせにはならない。結果を表示するだけなら軽い処理である。グラフィックが16色というのはちょっと困りものだが，ファイル上にフルカラー，メモリ上に65536色データを持っておき，表示のみ誤差拡散をかけてやれば，そこそこの画質は出る。

それよりも大きなメリットがある。マルチタスク，マルチウィンドウならではのメリットである。描画しながらそれを横目に形状データをエディタで編集することができる。現段階では，C−TRACEからエディタをチャイルドプロセスで呼び出し，形状データの編集が終わったらエディタを抜けてC−TRACEへ戻り，……ということを繰り返さなくてはならない。意外にこれが不快なのだ。SX−WINDOWならば，そこから抜けずにすべての作業が済ませられる。ちょっと長い計算に入ったら原稿を書きながら待つ。試し計算が終わるまでの時間ならピンボールもいい。

SX−WINDOWがようやく使いものになってきたいまが旬である。トランスピュータ版だけでいいから，SX−WINDOW上で動くレイトレーシングソフトウェアを作ってみてはどうだろうか。快適な環境になること受けあいである。簡単なモデラーでもついていればもっといいのだが。

これがメタビューだ

作業のようす。上がワイヤビュー

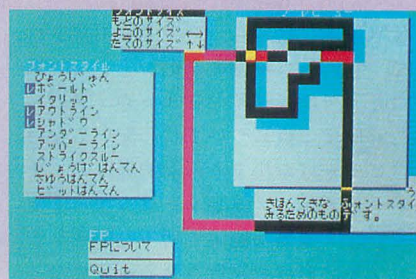
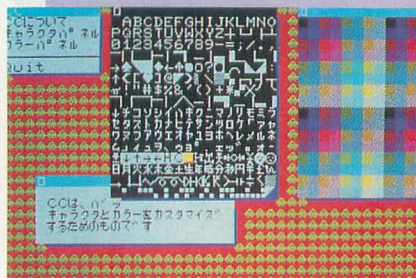
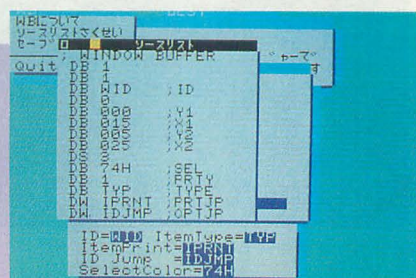
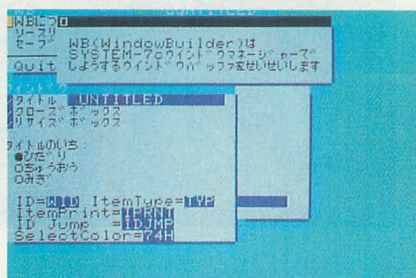
「伝説の男」が再び現れた。今度はマルチウィンドウシステムを引っ提げて。これでMacintoshの環境がMZ-700でも味わえる(かもしれない)。はたしてMZ-700に限界はあるのだろうか?

●

確かX68000用にSX-WINDOWが発表されるちょっと前くらいではなかったかと思う。「MacintoshのToolboxのようなものを作ろうと思うんですが……」と電話でもちかけられたことがある。冗談半分に「MZ-700で?」と聞くと予想どおり「ええ,そうなんです」と答えが返ってきた。

すでにおわかりの方も多いと思うが,古籏一浩君の登場だ。新しい読者の方のために解説すると,彼は16歳のときMZ-700版ゼビウスを完成して以来,本誌ではMZの名物男として知られている。MZ-700/1500/2500,X68000ユーザーでPC-9801やMacintoshも使っている。それでメインマシンがMZ-700というのはちょっとほかにはいない。

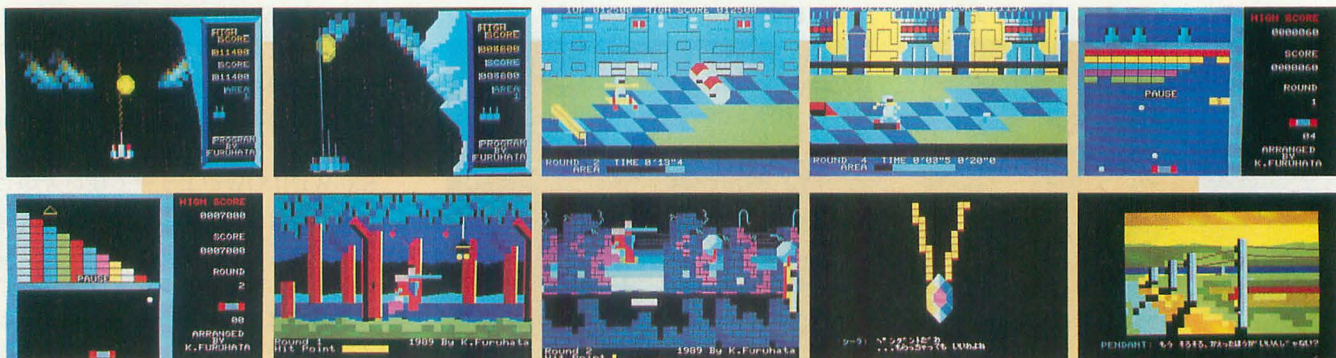
オリジナルシューティングゲームのSPACE BLUSTERシリーズを初め,アドベンチャー風紙芝居のEyelarth,そしてスペースハリアーをいずれもMZ-700上で実現し誌上に発表している。そして,その彼の技術の集大成ともいえるものが,SYSTEM-7Bという高機能ゲームパッケージだ。

MacintoshのToolboxはMacintoshシステムの要となる256Kバイトものシステムルーチン集。SX-WINDOWでいえばFSX.Xに相当するものだ。Z80マシンで,それもメモリの少ないMZ-700ではたして実現できるのだろうか? グラフィックがない分だけ有利そうに思えるが,SYSTEM-7Cでは「本体がグラフィックを持っていないのにビットマップの処理をサポートしている」のだ。とはいえ,SYSTEM-7Bを核にすればQuickdraw程度のものは簡単にできそうだし,なによりMZ-700はスペースハリアーをオンメモリで動かしたという実績もある。うーむ。

さらにSYSTEM-7Bの後継である以上,その上でリアルタイムゲームが動くのは間違いない。マルチウィンドウだから遅い,などという常識に甘えたウィンドウシステムとは一線を画すことが容易に予想される。

結論がここで挙げた写真である。フォントプレビュア,ウィンドウビルダ,背景変更ユーティリティなど。まだ大きなアプリケーションは動いていないが,カセットテープベースでちゃんとマルチウィンドウが実現されているのだ。

これらのプログラムがサークルEXTRAから配布されることになった。SYSTEM-7C関係のソフトウェアのみでなく,SYSTEM-7Bを使って作られたものや,その他のプログラムなどもまとめて配布される。下にある写真はその一部。どこかで見たようなものもあるが,画面にモザイクがかかっててよくわからない。たぶん気のせいだろう。

*　　*　　*

さて,もうひとりの伝説の男,横内威至君のグラディウスも完成間近! もしかしたら配布されるかもしれないのでX1turboユーザーは気長に待つように。

SYSTEM-7C

古籏 一浩

新世代マルチウィンドウシステム，といってもX68000ではありません。MZ-700です。機種に依存する部分はほとんどないということなので，Z80機種ユーザーで移植したいというツワモノはEXTRAに連絡してみてください。

MZ-700/1500ユーザーの方々お待たせいたしました，SYSTEM-7C Ver1.0の発表です。久しくOh!Xに載らなかったので，どうしたのかな，という人もいるでしょう。実は長いあいだ，今回発表するSYSTEM-7Cを作成していたのです。昔System-7Bというシステムを発表したのですが，最近の読者の方は知らないと思いますので，少し説明させてもらいます。System-7BはMZ-700/1500用ゲームシステムとしてOh!Xに掲載されました。わずか4Kバイトという大きさですが，各種PUT(オーバー付き，拡大縮小付き)やグラフィック，特殊効果，サンプリング，判定ルーチン，メニュー選択などを備えていました。

作った本人にとってSystem-7Bは大変ゲーム開発に効果を示してくれました。従来の半分以下の期間でゲームが作成できるようになり，またほかのところへ力を入れることができるようになったからです。しかし，当然のことながら不満もありました。特に，ゲームで使用されないルーチンがたくさんあるのです。また，ゲームを作成する以前にツールがほとんどありませんでした。その他，スピードの限界(なるべくコンパクトにするために多少処理スピードを犠牲にした)もありました。ならば改良すれば？　という意見もあることでしょう。しかし，ジャスト4Kバイトで作ってしまったために改良はほとんど無理でした。そういったことをふまえて，今度はもっとよいものをと考えて作ったのがSYSTEM-7Cです。

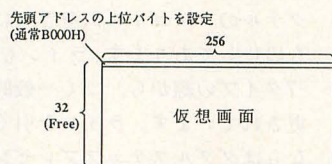
System-7Bからの改良点

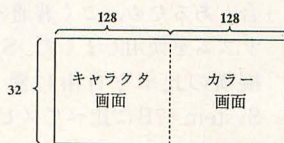
System-7Bの欠点を克服するために多少工夫をこらしました。まず，仮想画面の構成です。System-7Bは64×32という大きさでしたがSYSTEM-7Cは256×自由(ただしメモリの許すかぎり)としました(図1を参照してください)。また，仮想画面に使用できるメモリアドレスも256バイト単位でほぼ自由に設定できます。この構成はZ80にとって究極に近い仮想画面の構成です。この仮想画面の改良により，System-7Bよりもさらに高速，コンパクトとなりました。

仮想画面でもうひとつ改良されたことがあります。従来はキャラクタ画面とカラー画面が別々になっていましたが，SYSTEM-7Cではその区別がありません。つまり自由に設定可能です。全部キャラクタ画面として使用してもよいし，適当に割り振って別々に使用してくれてもかまいません。ただし，フォントマネージャ以降のマネージャに関しては，X座標が128以上の場合はカラー画面として使用しています。

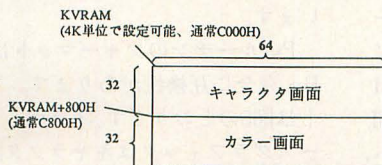
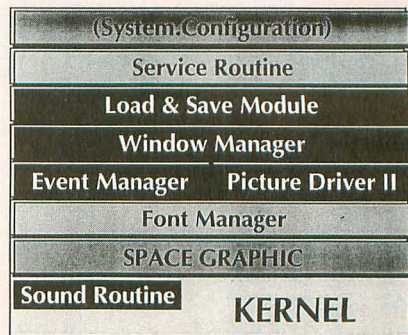
次に大きな変更として，システムの階層化があります。図2のように階層化されており，カーネル，スペースグラフィック，フォントマネージャ，イベントマネージャ，ウィンドウマネージャなどがあります。これにともない，従来は直接サブルーチンを呼び出していたのを，間接的に呼び出すよ

図1

●SYSTEM-7cの仮想画面構成　(カーネル&スペースグラフィック)

★高速アドレス指定方法
(Hレジスタ=X座標、Lレジスタ=Y座標)
X=X+1　：INC H
Y=Y+1　：INC L

●SYSTEM-7cの仮想画面構成　(マネージャー関係)

★高速アドレス変換方法
(Hレジスタ=X座標)
キャラクタ画面に切り替え　：RES 7,H
カラー画面に切り替え　：SET 7,H

●System-7Bの仮想画面構成

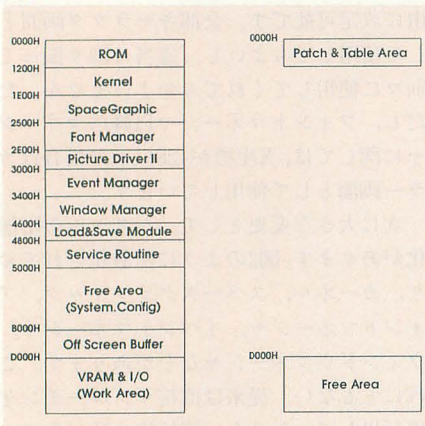
図2

うに変更しました。これでバグや不都合があっても修正が容易にできます。スピードが気になる人にいっておきますが，このようにしてもSystem-7Bより高速です(別にSystem-7Bが遅いということではありません，念のため)。

システムの階層化にともない，SYSTEM-7Cのアドレスは固定となっています。System-7Bはアドレスコンバータで，2000H～C000Hに4Kバイト単位で移動させることができました。アドレスが半固定だったために，発表当初は9000H版だったのがのちのソフトなどはほとんどがB000H版だったという面倒なことが起きてしまいました。SYSTEM-7Cは面倒なことはやめてさっさと固定アドレスにしてしまいました。でも，固定アドレスにすると，いろいろな問題が出てきそうな気がしますねぇ。

最後に少しの改良ではありますが，入力パラメータをなるべく同じにしてあります。特にグラフィック描画ルーチン，PUTルーチンはこの改良がなされています。

大きな変更点としては以上です。

SYSTEM-7Cのメモリマップを図3に示します。フリーエリアは5000Hからです。

図3

時代の波

時間というものは刻一刻と過ぎていきます。読者のなかにはMZ-700/1500などという機種名は聞いたこともない，そんな人がいるでしょう。時代の流れですね。というわけで，とりあえずMZ-700/1500の簡単なスペックを載せましょう。

CPU：Z-80Aクロック約3.58MHz（X68000は10MHz）
RAM：64K　(1Mの16分の1)
テキスト：40文字×20行固定
グラフィック：できません（MZ-1500は320×200の8色表示）
サウンド：短形波で単音のみ（MZ-1500はPSG 6重和音）
メディア：カセットテープ(MZ-1500はQD)，FDも一応使えます。

D000H以降のD-RAMも使用可能ですが，仮想画面としては使用できません。SYSTEM-7Cの仮想画面として設定できるのは1200HからCF00Hまでです。また，0000Hから0FFFHはマネージャ群が使用していますのでゲーム以外で使用しないでください(バグ発生時のパッチ領域)。

では各ブロックの説明をしましょう。

カーネル

これがないと話になりません。基本的な処理を受け持ちます。やっていることは以下のようなことです。

敵弾の発射/移動(Lineと共用)
一時停止
ゲームキー入力
自機移動
判定
乱数
演算ルーチン
スコア加算表示
カラー変換
(キャラクタ移動システム)
(サウンドルーチン)
(割り込み処理)
(データ圧縮展開)

カーネルのルーチンはSystem-7Bでよく使用されたものを改良したものがほとんどです。このなかでスコア表示ルーチンが多少バージョンアップされています。従来は“00001234”のように必ず先頭にゼロがつきましたが，SYSTEM-7Cではゼロサプレスすることができます。その場合，左詰め，右詰め表示が選択できます。スコア加算ルーチンもSystem-7Bは一度に9999しか加算できませんでしたが，今回は99999999まで一度に加算できます。

ジョイスティック関係はシャープ提供のルーチンを改良し，一度にすべての入力を行っています。このため，シャープ提供のものよりも約2倍速くなっていますが，その分精度が悪くなっています。乱数ルーチンと演算ルーチンは筑紫さんに作っていただきました。括弧内は入ってはいますが未公開です。

サウンドルーチンはSoundRoutine Ver.9.0縮小版が搭載されています(Ver.2では別物になると思います)。圧縮展開ルーチンは以前掲載されたものとほとんど同じです。割り込み処理は，4つまで可能です(1つはサウンドルーチン用)。ほかにも未公開ルーチンとしてはキャラクタ移動システム，敵表示，敵判定ルーチンもあります。ROMモニタを使用するのを避けていますので，ジャンプテーブルを含めて2Kバイトと若干大きめです(SoundRoutineを除く)。

スペースグラフィック

グラフィック描画部分にはスペースグラフィックというたいそうな名前がついています。まあ，グラフマンよりはまともだとは思いますが。MacintoshはQuickDraw，NeXTはDisplay Post Scriptが搭載されていますね。スペースグラフィックの機能は以下のとおりです(図4参照)。

Put（ノーマル，ハイパー，オーバー，独立変倍拡大/縮小）
Get（グラフィック取り込み）
ライン
ボックス/ボックスフィル
サークル/サークルフィル(正円のみ)
ポリゴン/ポリゴンフィル
スクリーンエフェクト
リングスクロール(4方向)
ペンスタイルで描画

リージョン(領域)，扇形，角の丸い四角形はありません。角の丸い四角形はシステムコールの93，94に隠れてはいますが，あまり使い道がないでしょう。もともとダイアログボックスを表示させるためにつけたのですが，用なしになってしまいました。Put, Get，リングスクロール以外のルーチンはさまざまなペンスタイルで描画可能です。ペンサイズも自由です。

System-7Bではサークルは8分割アルゴリズムでしたが，今回はOh!Xのマシン語カクテルのアルゴリズムをさらに高速化したものにしてあります。ラインも同様に，FM-7タイプの線から，ごく一般的なものに変更されています。ラインを引くだけの処理ならばダブルステッププレゼンハムのアルゴリズムを使えば相当の高速化が可能ですが，ゲームで使用する敵弾の移動などの都合もあるため，ごく普通のラインのアルゴリズムを使用しました。SYSTEM-7Cでは縦横の比率を自由に変えられますが，System -7Bに比べてスピードは遅くなっています。あとポリゴンフィル(多角形塗りつぶし)ですが，これは以前Oh!FMに掲載されたアルゴリズムを多少改良し搭載しています。

Putルーチンのフォーマットは System-7Bと完全に互換性があります。フォーマットは図5のとおりです。注意点としてはスペースグラフィックはキャラクタ画面とカラー画面とを区別なく描画してしまいますの

で，System-7BのPutと同様の処理を行うためには，Put.ChrとPut.Atbを組み合わせて使用してください。

プログラムサイズは約2Kバイトです。

フォントマネージャ

その名のごとく文字を扱います。

- キャラクタ/ビットマップフォント
- 横書き(左から右，右から左)，縦書き(上から下)
- 左詰め(上詰め)，右詰め(下詰め)，センタリング
- ビットマップフォント時，各種スタイル設定可
- ビットマップフォントは縦横自由にサイズ変更可

フォントマネージャができることはだいたい上記のようなことです。MZ-700でありながらなぜかビットマップフォントが扱え，図6にあるようなスタイルが設定でき，マルチフォントでなおかつ縦書きもサポートしています。左詰め(上詰め)，右詰め(下詰め)，センタリングも可能です。ビットマップフォントの字母(もとの大きさ)は，最大16×16ドット以内です。従来のキャラクタ文字も使用できます。また，図7のようにデスティネーションレクタングルとビューレクタングルがあり，フォントマネージャはビューレクタングルからはみ出した部分は描画しません。フォントマネージャキャラクタ文字の場合，仮想画面だけでなくVRAMにも直接表示できます。

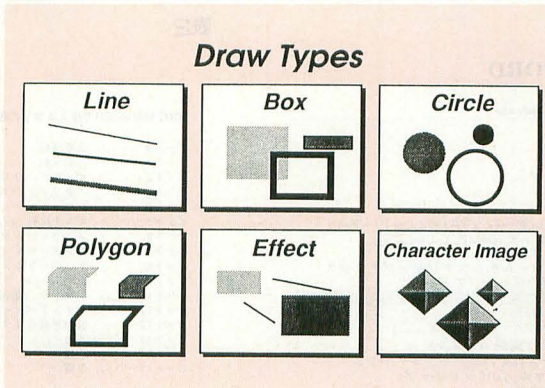
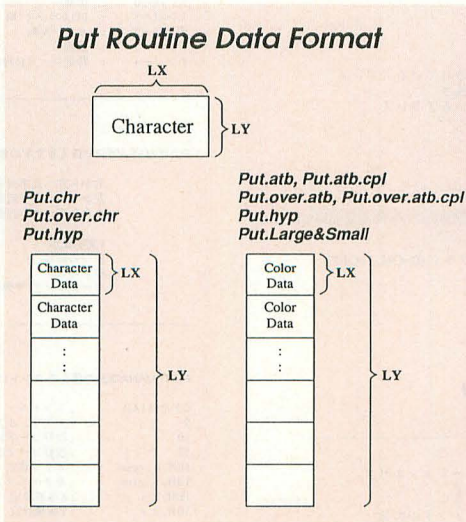
フォントマネージャはあらかじめSET.FONT.RECORDによってフォントレコードを設定しておく必要があります。フォントレコードは，ショートとロングの2種類があります。ロングのほうはすべての項目を設定することができますがショートのほうは使用文字がキャラクタであることやフォントサイズその他が固定されています。キャラクタ文字しか扱わないならばショートを使用したほうが楽です。MZ-700にとってはかなり無茶苦茶なマネージャともいわれています。X68000だったらよかったかもしれませんね。プログラムサイズは約3Kバイト弱の大きさです。

それでは，フォントレコードの解説をしましょう。

●ロングフォントレコード

ロングフォントレコードはフォントマネージャが参照する内容をすべて設定することができます。順番に上のほうから解説していきましょう。まずバージョンですが，これは現在のバージョン，つまり“1”にしてください。次にX，Y座標ですが，Y座標が1バイトに対しX座標は2バイトになっています。これは当初，セミグラフィックで表示させようかなと思っていたのですが，都合によりやめてしまったのです。そのときの名残りです，はい。たとえばXY座標をX=67，Y=12に設定したいときは次のようにしてください。

```
DB 12   ；Y座標設定
DW 67   ；X座標設定
```

次にデスティネーションエリアとビューエリアです。フォントマネージャはテキストがデスティネーションエリアに収まるように調整します。テキストはディスティネーションエリアの左上(右から左書きと縦書きのときは右上)から始まって，右端(右から左書きのときは，左端，縦書きのときは下端)にくるたびに折り返されます。テキストが下端(縦書きのときは左端)を越えるとスクロールアップ要求がフォントマネージャからアプリケーションに伝えられます。スクロールする必要がある場合はアプリケーション側で処理してください。ビューエリアは，デスティネーションエリア中の実際に表示される部分を設定します。注意点として，デスティネーションエリアの横幅よりも大きな文字を表示させるとハングアップする可能性があります。

VRAM.SWは表示画面を仮想画面とするかVRAMとするかのスイッチです。“0”を設定すると仮想画面，“1”を設定するとVRAMに表示するようになります。ただし，この設定はビットマップフォントに対しては無効になります。つまり，キャラクタ文字だけが直接VRAMに出力することができます。

TATEは，“0”=横書き(左から右)，“1”=横書き(右から左)，“2”=縦書き(上から下)のいずれかの表示方法が設定できます。通常は“0”の左から右に設定しておけば差しつかえないでしょう。右から左書きというのは昔の日本では使われていましたが，今は使い道がないと思います。この場合，濁音などの処理はしていませんので，『ギク』を表示させると『グキ』となります。

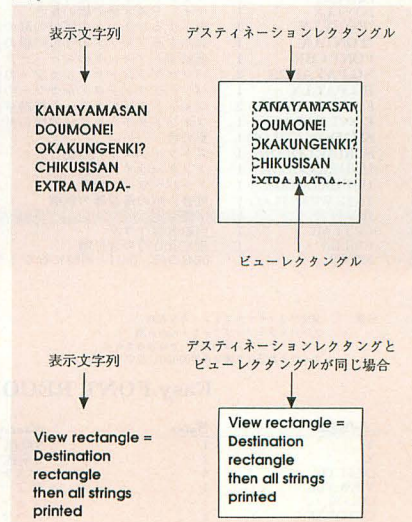
縦書きは，ごくノーマルに上から下へ表示していきます。縦書きの場合は濁音など

図4

図6

上記に示したスタイルの組み合わせ以外にフォントタイルとバックグランドタイルが使用できます。

図5

図7

の処理方法が，TATE.WRAP.FLAGによって設定できます。“0”だと濁音などの処理は行わず，“1”にすると濁音などのコードが見つかった場合，右側に表示します。したがって，TATE.WRAP.FLAG＝“1”の場合は，2行使用することになります。

次のMZ.ASCは“0”で従来のMZ-700のASCIIコードになり，“1”を設定するとユーザー定義のASCIIコードが使用されるようになります。ユーザーASCIIコードが設定された場合，次に続くCONTROL.ADRSとCONTROL?.ADRSを設定しなければなりません。CONTROL.ADRSはコントロールコードがきた場合のジャンプテーブルのアドレスです。CONTROL?.ADRSはコントロールコードかどうかチェックするサブルーチンのアドレスを設定します。サブルーチン内ではAレジスタ以外はなるべく破壊しないようにしてください。このサブルーチンからリターンしたときに，キャリフラグ＝“1”(キャリ)であればコントロールコードとみなされ，先ほどCONTROL.ADRSで設定されたテーブルを参照してジャンプします。テーブルはASCIIコード00_Hから順番にアドレスが設定されていなければいけません。

SCRL.JPは先ほど説明したのでパス。TAB.TBLは予備です。

PROP.FLAGとPROP.TBLはビットマップフォント設定時のみ有効になります。PROP.FLAGはその名のごとく，“1”のときプロポーショナルになります。“0”にすると設定した字母の幅で座標が進んでいきます。PROP.TBLはPROP.FLAG＝“1”のときに参照されるテーブルで，ASCIIコード00_Hから順番に，ofstX, ofstY(それぞれ1バイト)の順番で設定されている必要があります。ofstX, ofstYは現在のX, Y座標に加算すべき値です。負の値も使えますが，表示方向が逆になったりしてまぎらわしいので，なるべく正の値を使ってください。

ASC.TBLですが，通常$0A92となっています。これはROMに入っているASC→DISPLAY変換テーブルを使っているからです。オリジナルにしたい場合はこのアドレスを，変更してください。このテーブルもASCIIコード00_Hから順番に，変換されたディスプレイコードが格納されている必要があります。

次のFONT.TBLですが，これはビットマップフォントが指定されているときのみ有効になります。このテーブルもASCIIコード00_Hから順番にそれぞれのビットマップフォントの格納されているアドレスがならんでいる必要があります。

FONT.TYPEはフォントを指定します。“0”のときキャラクタ，“1”のときビットマップフォントになります。

FONT.STYLEはビットマップフォントの書体を設定します(キャラクタの場合は無効)。それぞれのビットが“1”のとき，ボールド，イタリックなどを指定したことになります。

COLORはフォントの色です。これはビットマップフォント/キャラクタ双方に有効です。

COLOR.EFはキャラクタ文字のときのみ有効です。これはビットでなく，数値になっています。

FONT.LYとFONT.LXはフォントの字母を設定します。これはビットマップフォント/キャラクタ文字双方に有効です。キャラクタ文字でFONT.LX＝2, FONT LY＝3を設定した場合，横2文字単位で表示され，改行されたときは3行ずつになります。

表1

FONT RECORD

LABEL	Bytes	Contents
VERSION	1	バージョン
Y	1	Y座標
X	2	X座標
DEST.TBL	4	デスティネーションエリア
VIEW.TBL	4	ビューエリア
VRAM.SW	1	出力画面スイッチ
TATE	1	縦書き、横書きフラグ (0=左→右, 1=右→左, 2=上→下)
MZ.ASC	1	MZ用アスキーコードスイッチ (0=MZ, 1=User設定)
CONTROL.ADRS	2	MZ.ASC>0の時のジャンプアドレス
CONTROL?.ADRS	2	MZ.ASC>0の時のコントロールチェックジャンプアドレス
SCRL.JP	2	スクロールアップ要求時のジャンプアドレス
TAB.TBL	2	タブテーブルのアドレス
PROP.FLAG	**1**	プロポーショナルのON, OFF
PROP.TBL	**2**	プロポーショナル指定時のテーブル (X, Y, ...)
ASC.TBL	*2*	*通常$0A92 (ASC->DISPLAY変換テーブル)*
FONT.TBL	**2**	フォントアドレスが格納されているテーブルアドレス
FONT.TYPE	1	フォントタイプ (0=ノーマル, 1=ビットマップ)
FONT.STYLE	**2**	ビットマップフォントの書体
COLOR	1	フォントカラー
COLOR.EF	*1*	*ノーマルフォントの書体*
FONT.LY	1	フォントの字母の縦の長さ
FONT.LX	1	フォントの字母の横の長さ
?FONT.LY	**1**	表示するフォントの字母の縦の長さ
?FONT.LX	**1**	表示するフォントの字母の横の長さ
FONT.CHR	**1**	表示するフォントのキャラクタ
BG.PAT.ADRS	**2**	バックグランドタイルカラーの格納されているアドレス
BG.PAT.LX	**1**	バックグランドタイルカラーの横の長さ
FONT.PAT.ADRS	**2**	フォントタイルカラーの格納されているアドレス
FONT.PAT.LX	**1**	フォントタイルカラーの横の長さ
SHADOW.COLOR	**1**	影の色
STRIKE.COLOR	**1**	ストライクスルーの色
UNDER.COLOR	**1**	アンダーラインの色
UPPER.COLOR	**1**	アッパーラインの色
TATE.WRAP.FLAG	1	縦書き時の濁音等の制御
JIKAN	1	字間設定
CR.FLAG	1	自動改行フラグ
CR.LEN	1	指定改行時の改行幅
MZ.CR	1	0DHの時、0AHも同時に行なうかのフラグ (0=ON, 1=OFF)

注意:　袋文字はノーマルフォントのみ有効
太字はビットマップフォントのみ有効
フォントレコードは内部ワークは含みません
アミカケ部分は予備なので0000Hに設定してください

Easy.FONT RECORD

LABEL	Bytes	Contents
Y	1	Y座標
X	2	X座標
DEST.TBL	4	デスティネーションエリア
VIEW.TBL	4	ビューエリア
COLOR	1	フォントカラー
COLOR.EF	1	ノーマルフォントの書体

表2

FONT MANAGER

FONT MANAGERで扱える文字の種類(ただし、BIT MAP FONTのみ)

ビット0	:	上下反転
ビット1	:	左右反転
ビット2	:	ボールド (太字)
ビット3	:	イタリック (斜体)
ビット4	:	アウトライン
ビット5	:	ビット反転 (ここからはカラーバッファを操作する)
ビット6	:	ストライクスルー (中心線)
ビット7	:	アンダーライン (下線)
ビット8	:	アッパーライン (上線)
ビット9	:	BGパターン
ビット10	:	シャドウ (影付き)
ビット11	:	フォントパターン
ビット12	:	仮想画面のカラーのビット3が1の時には書き込まない
ビット13	:	予備
ビット14	:	予備
ビット15	:	予備

FONT MANAGERで扱える文字の種類(ただし、NORMAL FONTのみ)

ナンバー0	:	標準
ナンバー1	:	BGカラーとOR
ナンバー2	:	カラー反転
ナンバー3	:	透明
ナンバー4	:	特定コードに対してカラービット7を1にする

FONT SCRIPT

FONT MANAGERで扱える文字の表示方向、表示形態

右から左へ文字列を表示
左から右へ文字列を表示
上から下へ文字列を表示

1文字表示
左づめ表示
右づめ表示
センタリング(中央寄せ)表示

Control Code

FONT MANAGERで扱えるコントロールコード

0または1AH	:エンドコード
5	:大文字、小文字切り替え
10	:改行コード1
13	:改行コード2
1BH, 0, color	:カラー設定
1BH, 1, effect	:カラーエフェクト設定 (ただし、ノーマルフォントのみ)
1BH, 2, x	:X座標設定
1BH, 2, y	:Y座標設定

?FONT.LY, ?FONT.LXはビットマップフォントの場合のみ有効です。これらは実際に表示される大きさを指定します。縦横独立に指定できます。もし，プロポーショナルがONのときは自動的に文字幅(改行幅)が計算されます。プロポーショナルがOFFだった場合はここで設定された表示サイズが文字幅(改行幅)になります。

FONT.CHRもビットマップフォントのときのみ有効です。ビットマップフォント表示時，キャラクタ画面に書き込むディスプレイコードを設定します。通常$5A_H$です。

BG.PAT.ADRS, BG.PAT.LX, FONT.PAT.ADRS, FONT.PAT.LXはビットマップフォントで，なおかつバックグランドタイル(フォントタイル)が指定されたときに有効になります。~.ADRSは1バイトのカラーコードが格納されているアドレスを設定します。そのタイルパターンの横幅が~.LXになります。

SHADOW.COLOR, STRIKE.COLOR, UNDER.COLOR, UPPER.COLORもビットマップフォントで，なおかつその指定がされたときに有効になります。これらはすべて1バイトのカラーコードです。

TATE.WRAP.FLAGはすでに説明しました。JIKANは予備です。CR.FLAG, CR.LEN, MZ.CRは“0”を設定してください。

●ショートフォントレコード

ショートフォントレコードは，キャラクタ文字のみを使用したい場合に有効です。ロングフォントレコードと違い，XY座標，デスティネーションエリア，ビューエリア，フォントカラー，書体しか設定することはできません。それぞれの内容は，ロングフォントレコードとまったく同じです。

イベントマネージャ

イベントマネージャという名前はついていますが，キー，ジョイスティックなどの入力を管理するくらいです。イベントキューは用意していません。また，SYSTEM-7Cではなるべくダブルクリックを排除しています。あのHyperCardもダブルクリックは避けていますし。イベントマネージャにはシングルクリック，移動などのイベントチェックルーチンが入っていますので，キー，ジョイスティックの設定による違いはここで吸収されます。つまり，GET.CKICK.STATUSを呼び出すだけで簡単に入力状態をチェックすることができます。

あと，イベントマネージャには1行のテキストエディットルーチンも含まれています(非公開，ウィンドウマネージャの下請けルーチン)。イベントマネージャもフォントマネージャ同様にレコード設定が可能ですが，イベントマネージャだけはすでに設定されているので，特に設定しなおす必要はありません。イベントレコードについては表2を参照してください。

プログラムサイズは約1Kバイトです。

ウィンドウマネージャ

世の中にはたくさんのウィンドウシステムがあります。MacintoshのウィンドウシステムNeXTのNeXT Step(?), IBM PCのMS-Windows, X68000のSX-WINDOW, 同じくKo-Window, UNIXのX-Window(ほかにもあります), AmigaのWorkBench, AtariのGEM(PC-9801にもあります)などなど。が，SYSTEM-7CのウィンドウシステムはSX-WINDOWなどとはかなり異なっています。だいたいメモリが64KバイトしかないMZにほかと同じようなシステムを搭載したらすぐにメモリが足りなくなってしまいます。また，新しいプログラミングスタイルを勉強するのも面倒です。そこで次のようにして，メモリの消費やプログラミングの手間を少なくしました。

●基本的にオフスクリーンバッファ[*1]を持たない

つまり，MacintoshやSX-WINDOWと同じです。基本的にNeXTなんかではウィンドウひとつにつきオフスクリーンバッファを持っていますが[*2]，そうすると大量のメモリが必要です。そこでMacintoshを参考にアクティベートイベントが発生したら，プログラムによってウィンドウを書き換えることにしました。こうすればオフスクリーンバッファを持たない分だけメモリが節約できます。ただし，ペイントタイプのソフトはオフスクリーンバッファを持つべきでしょう。その場合の表示ルーチンは自前で作成する必要があります。

普通オフスクリーンバッファを持たないシステムはウィンドウの書き換えが面倒です。SX-WINDOWも面倒な感じです。SYSTEM-7Cもオフスクリーンバッファを持っていませんが，書き換え(アップデート)に関してはウィンドウマネージャが自動的に行います。つまりプログラマの負担がほとんどないように設計されています。だいたい，作った本人が面倒なことは嫌いだからという理由もありますが。

＊1 画面に表示されない画面，つまり仮想画面のこと。Macintoshではペイントソフトでよく使われている。

＊2 NeXTでは，buffered(オフスクリーンバッファあり)，retained(ウィンドウに隠される部分をバッファに保存)，none(オフスクリーンバッファなし)の3タイプが選択できます。Amigaなども同様です。

●ウィンドウもメニューもまったく同じものとする

これはNeXTマシンを参考にしました。あれってメニューとウィンドウが似ているでしょう? ということでメニューを選ぶのも，ウィンドウの中の項目を選ぶのも同じこととしました。メニューマネージャがいらない分だけメモリが節約できます。

ところで，メニューの形式(図8参照)には何種類かありますが，上記のような理由により，NeXTと似たようなメニュー(フローティングメニュー?[*3])となっています。ポップアップメニュー[*4]にするとまた別の処理が必要だし，プルダウンメニュー[*5]&ティアオフメニュー[*6]も同様です。メモリにもう少し余裕があればポップアップメニューはつけるべきでしょうね。ちなみにSYSTEM-7Cでは階層化メニューはサポートしません(する予定もない)。あとキーボードショートカット[*7]はサポートしておきました。

表3

入手方法

現在，System-7Bを使ったものは誌上で2つしか発表されていませんが，以前予告したもののなかで，Galaxy ForceⅡ以外はすべてありますのでほしい人はEXTRAへ注文してください。グラフィックツールや未公開ゲームもあります。QDはフォーマット済みのものを送ってください。

住所を書いた紙も同封してもらえるとありがたいです。ついでに送料も負担していただけるとうれしい。

ちなみにSYSTEM-7CのゲームはOh！Xには掲載されない場合（著作権等々）がありますので，なるべくEXTRAに入会したほうがゲームが簡単に入手できるというメリットがあります。ただし，Oh！X掲載が確定したものは，掲載したものは，掲載後に配付になります。

〒811-42 福岡県遠賀郡岡垣町戸切794-3
筑紫 高宏

＊3　常に画面上に表示されているようなメニュー。なんとなく浮いているように見えるからこんな名前なんでしょうか。ほとんどティアオフメニューみたいなものです。

＊4　SX-WINDOWのメニューのように右ボタンを押すと，わいて出てくるメニュー。

＊5　Macintoshのように常にメニューバーが表示されているメニューで，メニューを選択すると，その下に選択項目が出てくるメニュー。

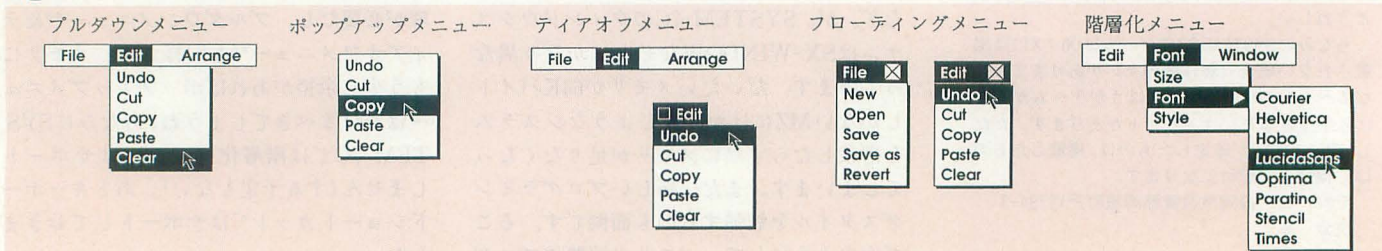
＊6　MacintoshのHyperCardのように，メニューバーから切り離して画面上においておくことができるメニュー。参考までにMacintoshでは上記すべてのメニュー形式が可能です。ほかにも世の中には変わったメニューが存在します。

＊7　MacintoshやNeXTでは，Commandキーとほかのキーの組み合わせでメニューを選択するのと同様な処理ができます。たいていのウィンドウシステムでは可能です。

●ウィンドウにあるものはすべてアイテム(オブジェクト)とする

タイトルバーもクローズボックスもただの1アイテムにすぎないということです。このためウィンドウの形状はたったひとつしかありません。つまり，アラートボックス，×××付きウィンドウ，ドキュメントウィンドウという区別がありません。必要ならばつけるといった感じです。そのため，SX-WINDOWと違いダイアログマネージャ，コントロールマネージャがありません。システムで定義されているアイテムには次のようなものがあります。

タイトルバー
クローズボックス
リサイズボックス
テキストエディット
区切り線
キャラクタパネル
カラーパネル
ラジオボタン
チェックマーク

アイテムはそれぞれ独立して機能するようになっています。したがってほとんどの場合，あるアイテムの機能を変更してもほかのアイテムには影響を及ぼしません。逆に，あるアイテムとほかのアイテムが連動するようなものは作りにくくなっています。でも，あとから簡単にアイテムの追加削除ができるし，その際ほかのアイテムに影響がないのは，とても便利だと思いますが。

●面倒なことはシステムが受け持つ

SX-WINDOWなどのようにアップデートイベントがうんぬんということはありません。そういうことはほとんどウィンドウマネージャが処理します。イベント管理もプログラムする必要はありません。プログラマはウィンドウレコードを設定し，ウィンドウマネージャのイベントループへジャンプするだけです(プログラム例参照)。

じゃあ，タイトルバーが押された場合などの処理とかはどうするんだ? という人もいますね。そういった処理もシステムが処理します。タイトルバーなどはシステムが管理するからいいけどユーザーアイテムはどうするのだ，という人もきっといるでしょう。先ほど，ウィンドウにあるものはすべてアイテムと書きました。タイトルバーもそのうちのひとつにすぎないし，ユーザーアイテムもそうです。

ということで，アイテム1つひとつにつき，選択された(押された)ときの処理をプログラムしておけばいいわけです(HyperCardでいうスクリプト)。タイトルバーなどは，たまたまシステムで記述してあるだけです。つまりアイテムが選択されたときの処理をプログラミングすればよいのです。ひどい話ですが，いままで使っていたサブルーチンをアイテムとして登録するだけで動きます。レジスタは保存する必要ないし，割り込み以外は特に制限はありません。

システムで定義されているアイテムのIDは負の値となっています。ユーザーアイテムのIDは正の値です。

※サンプルプログラム

```
XOR  A
LD   HL,WINDOW.RECORD
CALL SET.WINDOW.RECORD
LD   A,1
;イベントループのファンクション番号
JP   FUNCTION
```

●システムコール数が少ない

つまり，覚えるものが少ないわけです。逆にいえばないものは作らなければいけないということにもなります。ちなみにスクロールバーがありません。その他にもないものはたくさんあります。いずれ必要とあれば，サービスルーチンに組み込むようになるでしょう。

＊　　＊　　＊

だいたいこんな感じです。ウィンドウマネージャもあらかじめウィンドウレコードを設定しておかなければいけません。

ウィンドウの操作方法

操作方法の説明です。MacintoshやX68000とほとんど同じといってもウィンドウシステムなんて触ったこともない人のために用語とその説明をしましょう。その前にSYSTEM-7Cでは，図9のように操作するキー(またはジョイスティック)を設定することができます。普通はカーソルキーとZキー，シフトキーですが，次のプログラムにより設定を変更することができます。

```
KERNEL        EQU  $1200 ;または1200H
SETGAMEKEY    EQU  31*3+KERNEL
RIGHTKEY      EQU  0  ;カーソル，Z，SHIFTキー
LEFTKEY       EQU  1  ;W，A，D，X，?キー，シフトキー
JOY1          EQU  2  ;ジョイスティック1
JOY2          EQU  3  ;ジョイスティック2
LD            A,RIGHTKEY ;またはLEFTKEY，JOY1，JOY2
CALL          SETGAMEKEY
JP$00AD ;または00ADH
```

＊　　＊　　＊

さて説明に入りましょう。キーはカーソルキー，Zキー，SHIFTキーとしていますが，設定により変わりますので適当に読み替えてください。また，ウィンドウの名称については図10を参照してください。

まず，ウィンドウを動かすにはウィンドウの一番上(タイトルバー)にカーソルを移動させZキーを押したままにします。押したままにしてカーソルキーを押してみてください。カラー反転した枠が移動しますね。

図8

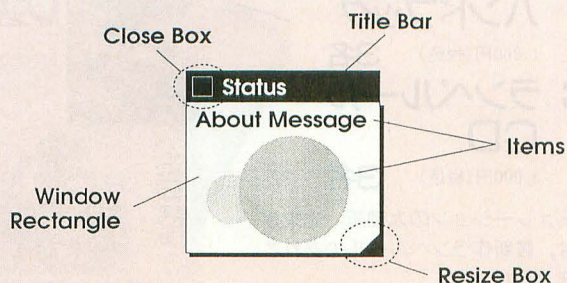
自分の好きな位置に移動させたらZキーを離してください。ウィンドウが移動させた位置に表示されます。このような操作をドラッグといいます。

では次にメニューを選択してみましょう。選択したいメニューへカーソルを移動させてください。Zキーを押します。メニューが点滅してなんらかの変化があると思います。このようにメニュー(アイテム)の上で，Zキーを押すことをクリックといいます。

ウィンドウによっては，ウィンドウの左端に四角(□)がついているものがあるでしょう。ここをクリックするとそのウィンドウを閉じる(画面から消す)*8ことができます。通常メインメニュー以外はクローズボックスがついています。

ウィンドウの右下に，リサイズボックス(三角形)がついているウィンドウはサイズを変更することができます。リサイズボックスの上までカーソルキーを移動させます。ドラッグして好みの大きさにしてください。ただし，大きさには制限があります。

ウィンドウがたくさん表示されると操作したいウィンドウが下に隠れてしまい見えなくなることがあります。そういう場合は上のウィンドウを移動させ，下に隠れたウィンドウが見えるようにします。下になっているウィンドウ上でクリックします。すると下にあったウィンドウが一番手前に表示されます。

ただし，下に隠れたウィンドウでもアイテムがある場合はそのアイテムが選択されますの注意してください。なにもないところかタイトルバー上でクリックするのが無難でしょう。通常一番手前にあるウィンドウはタイトバーが黒くなっています。タイトルバーが黒くなっているウィンドウのことをアクティブウィンドウといいます。下に隠れていてタイトルバーが水色になっているウィンドウのことをインアクティブウィンドウといいます。通常アクティブウィンドウ内が操作の対象ですが，インアクティブウィンドウでも表示されているアイテムがあれば，そのアイテムに対して操作を行うことが可能です。

アイテムのなかには押し続けていると変化するものがあります。押し続けることをプレスといいます。フォントプレビュアの矢印がそうです。この場合，押し続けているとフォントサイズが変わります。

SYSTEM-7C用のアプリケーションは通常図11のような構成になっています。画面には最低ひとつメニュー(メインメニュー)があり，そのメニューの先頭項目は“～について”,最終項目は“Quit”(終了のことです)となっています。“～について”を選択すると，ソフトについての簡単な説明がでます。“Quit”を選択するとMZ-700またはMZ-1500のモニタに戻ります。

説明がなくても，適当にクリックすればなんらかの反応があるはずです。反応がなければなにもないと考えていいでしょう。

基本的な操作はこんな感じです。なるべくシンプルにいこうと考えているので，操作はクリック，ドラッグ，プレスしかありません。ダブルクリックはやりにくいので，なるべく使用しないようになっています。

*8 SYSTEM-7Cのウィンドウはほかのスタックやヒープには作成されません。そのため，クローズしてもウィンドウ情報は残ったままになります。SYSTEM-7Cでウィンドウをクローズするということは画面から見えなくすると考えてください。詳しいことは来月のウィンドウマネージャの詳細を読んでください。

ソフトウェア

SX-WINDOWと似たもの同士なのです。つまりほとんどない！ とりあえず2月7日現在動作しているものは，

- Font Previewer
  フォントプレビュア
- Color Customizer
  バックグランドパターン設定
- Record Maker
  ウィンドウレコードの生成
- Demo
  スペースグラフィックのデモ
- Window Builder
  ウィンドウの生成

以上5つです。

WindowBuilder(WB)は，画面を見ながらウィンドウの位置，大きさ，システムアイテムの設定をするものです。設定を終えたらアセンブラのソースリストを生成します。WBは毎日のように改良されているので，ここで書いたものよりも，配布時の機能は増えているはずです。WBはNeXTに搭載されているインタフェイスビルダというものを参考にしました。もし，SYSTEM-7Cがマルチタスクでアセンブラが同時に走れば，アセンブルしてWBで連結し，即実行ということができるのですが。まあ，今後の課題ですね。

Record Makerもウィンドウレコードのソースリストとメインプログラムを生成します。メインプログラムまで作成してしまうところが，なんともいえませんね。ほかにもグラフィックツールとかいろいろ揃えたいとは思っています。System-7Bのときも同じように考えたけど，System-7B上では限界があります。その点，SYSTEM-7Cはアプリケーション作成アプリケーション?(WB)があるので，いろいろ作れるはずです。あと，ローマ字仮名変換もできればほしいですね。ゲームの予定ですが，全然未定です。なにを作るかも不明です。適当に期待していてください。

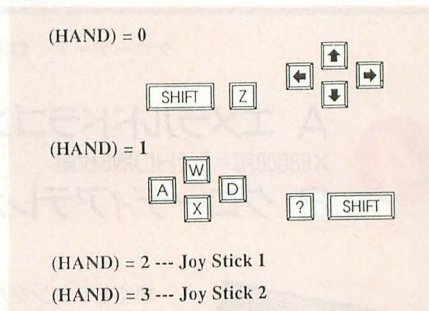
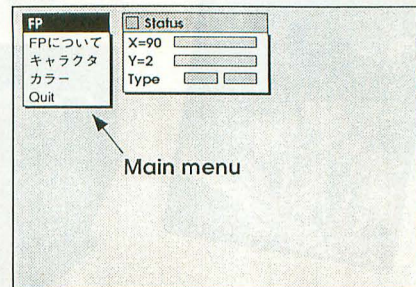
SYSTEM-7Cとその上で動作するプログラムのソースリスト，資料は一部の例外を除き，すべて公開します。来月は残りのマネージャその他について解説します。

図9

図10

Close Box
Title Bar
Status
About Message
Items
Window Rectangle
Resize Box

図11

愛読者特大プレゼント
創刊9周年記念

Oh!MZからスタートしてはや9年。いよいよOh!Xも，今年で10年目に突入しました。この1年間は，ディスクを3回もつけたし，まあ，10年目に向かう布石としては，こんなものでしょうと自負しています。とはいえ，月並みないい方ですが，やはりここまでこられたのは，応援してくれる読者の皆さんのおかげです。編集部一同，心を込めてお礼を申し上げます。でもって，これからもよろしくね。

さて，今年も祝！創刊9周年というわけで，たくさんのソフトハウスさんから，これまたたくさんのプレゼントをいただいています。もちろんこれは，全部皆さんにプレゼントしちゃうものですよ。ソフトどっさり，ソフト以外にも，ほかではなかなか手に入らないものなどもあります。

いつものことながら応募方法を熟読して（だって，ちゃんと書かない人多いんだもの）応募してください。お待ちしてます。

アートディンク　☎0472(79)9392

1

A 栄冠は君に　3名
X68000用　5"2HD版3枚組　9,500円(税別)

B 機甲師団　2名
X68000用　5"2HD版3枚組　9,500円(税別)

高校野球，戦争ものとタイプは違うが，采配をふるうのは同じこと。シミュレーションファンに。

イマジニア　☎03(3343)8911

2

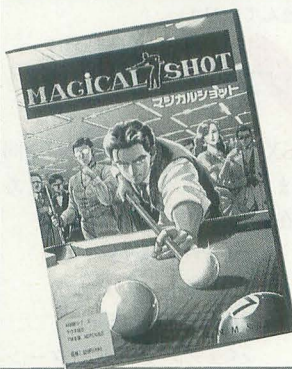
A シムシティー
X68000用　5"2HD版
9,800円(税別)　3名

B ポピュラス
X68000用　5"2HD版
9,800円(税別)　3名

イマジニアからは1990年度GAME OF THE YEARで，強さを見せつけたこの2作をプレゼント。

M.N.M Software　☎0423(60)3084

3

MAGICAL SHOT　5名
X68000用　5"2HD版　7,800円(税別)

3D表示によるリアルなビリヤードゲーム。マウスで簡単に操作できるのもうれしい。

グローディア　☎03(3220)5226

4

A エメラルドドラゴン　3名
X68000用　5"2HD版6枚組　9,800円(税別)

B グローディアテレカ　3名

早くもファンをがっちり摑んだグローディア。エメドラと特製テレカ，どっちを選ぶ？

光栄　☎045(561)6861

5

A ランペルール
X68000用　5"2HD版3枚組　9,800円(税別)　2名

B ランペルール ハンドブック
1,860円(税込)　3名

C ランペルール CD
3,000円(税込)　3名

シミュレーションの大御所，光栄からは，最新作ランペルールのあれこれをプレゼント。

コナミ ☎03(3264)5678

パロディウスだ!

X68000用 5"2HD版2枚組

9,800円(税別) 3名

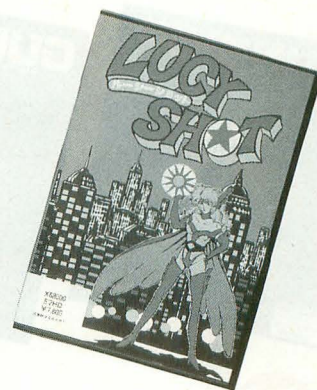
お馴染みのキャラクターが勢揃い。発売と同時に売り切れ! という人気ぶりのゲーム。

日本ソフテック ☎0425(82)1502

ルーシーショット

X68000用

5"2HD版2枚組

7,800円(税別) 3名

ソフテックピンボール第2弾。アメリカンなグラフィックが雰囲気モンでよい。

日本ファルコム ☎0425(27)6501

8

A ワンダラーズ・フロム・イース 3名

X68000用 5"2HD版4枚組 8,700円(税別)

B YsIII ステッカー 10名

C YsIII 5インチディスクホルダー 5名

D YsIII スポーツタオル 5名

E YsIII マウスマット 5名

F YsIII ペンセット 5名

今回はYsIIIシリーズで攻めてきたファルコム。いつもながらたくさんあってうれしいな。

ハミングバード ☎06(315)8255

9

A ラプラスの魔 1名

X1用 5"2D版2枚組 7,800円(税別)

B ラプラスの魔 1名

X68000用 5"2HD版3枚組 8,700円(税別)

C 5インチディスクホルダー&ロードス島戦記シール 5名

D ファイアボールテレカ 3名

X68000用, X1用と揃ったラプラスの魔。特にX1用は競争率が高いかもしれない!? ほかにディスクホルダーとテレカもあるでよ。

ボーステック ☎03(3708)4711

銀河英雄伝説II

X68000用 5"2HD版4枚組

9,800円(税別) 2名

発売から半年もたつのに, 依然人気の衰えないシミュレーションゲーム。MIDIにも対応。あたしはラインハルトが好きよ!

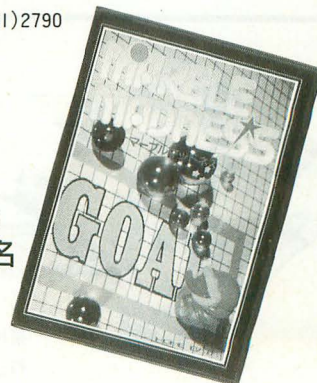
ホームデータ ☎078(261)2790

マーブル・マッドネス

X68000用 5"2HD版 9,700円(税別)

3名

ころころ玉を転がして時間内にゴールを目指す, ちょっと風変わりなアクションゲーム。ぜひトラックボールで。

マイクロプローズジャパン　☎0423(33)7781

12

A GUNSHIP　5名
X68000用　5"2HD版　14,800円(税別)
B GUNSHIPポスター　10名
C F-15ポスター　10名

マイクロプローズからは，またまたGUNSHIPをプレゼント。ほかにポスター2種もどうぞ。

HAL研究所　☎03(3252)5561

13

A ペーパーナイフ&ハサミセット
3名
B フラッグディスペンサー
3名

ファミコンでも有名なHAL研からは，“キレもの”セットとフラッグディスペンサーをプレゼント。

KDDクリエイティブ/ウッドブック
☎03(3207)5303

14

All in Noteの世界

5名

2,000円(税込)
ノートパソコン「All in Note」のすべてがわかる本。予備知識なしでも理解できる。

プレゼントの応募方法

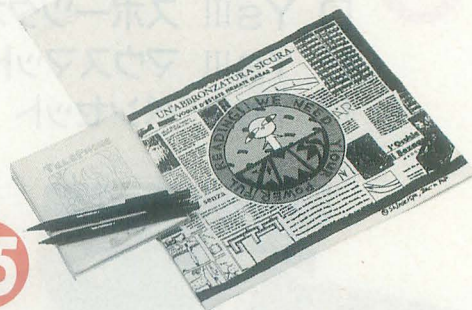
とじ込みのアンケートはがきの該当項目をすべてご記入のうえ，希望するプレゼント番号をはがき右下のスペースにひとつ記入してお申し込みください（同番号内に数種類ある場合は1-A，1-Bのように明記のこと)。締め切りは1991年6月18日の到着分まで。当選者の発表は1991年8月号で行います。

4月号プレゼント当選者

1中華大仙（北海道）太田哲雄（栃木県）毛塚健次（香川県）三島武典 2リングマスターII（茨城県）西方茂樹（埼玉県）八木秀訓 3エメラルドドラゴン（三重県）樋口潤次（大阪府）三木義雄 4ポピュラスCD（東京都）田代俊一（静岡県）堀江正樹 5サイトロンCD（大阪府）越智亮（福岡県）大津和之

以上の方々が当選されました。おめでとうございます。商品は順次発送いたしますが，入荷状況などにより遅れる場合もあります。また，公正取引委員会の告示により，このプレゼントに当選された方は，この号の他の懸賞には当選できない場合がありますのでご了承ください。

新声社　☎03(3293)9321

15

A テレカケース　10名
B オリジナルTシャツ　5名
C ボールペン&シャープペン　10名

スコルピウスが評判の新声社からは，ゲーメスト特製オリジナルグッズをプレゼントだ。

M 特別モニタ募集 HXD-140

X68000用　40Mバイト内蔵型ハードディスク
98,000円(税込)

1名

創刊9周年企画として，アイテムより内蔵型ハードディスクのモニタを大募集します。なお，モニタの方には，試用レポートを提出していただくことになりますので，あらかじめご了承ください。

アイテム　☎0466(27)1668

16

おみやげせっと

1名

某K氏が新婚旅行先で買った青ウミガメの肉煮込みと，パパイヤの味噌漬&粕漬です。

大人のためのX68000 [第8回]

[創刊9周年記念特別企画]

第2回 愛読者アンケート分析大会(その1)

Ogikubo Kei 荻窪 圭

今回の「大人のためのX68000」は創刊9周年記念特別企画として，3月号で行った「Oh!X愛読者特別アンケート」の結果を発表します。今回は（その1）ということで，おおまかで総合的なデータを発表します。詳細で関連的なデータは次号で。

「内緒話は嫌いだ。けど，人の内緒話を聞くやつはもっと嫌いだ」

「俺はお前のほうが，もっと嫌いだ」

なんなんだか。最近，吉田栄作がなんだかのたまっているシャープの盗聴防止コードレスホンのCMが耳について離れない，荻窪圭である。しかし，盗聴ねえ。電話の盗聴は罪だけど，無線の傍受はそうじゃない。盗聴と傍受の違いだが，傍受の場合，内容を他人に漏らした時点で罪になる。これは盗聴とはまた別である。怖いなあ。「盗聴」なんて言葉を安易に使ってしまうのが。「ラジオライフ」の味方するわけではないけど，コードレスホンを使うなら，無線機だということを自覚しなきゃ。聞かれるのがいやなら，微弱型コードレスにするとか，コードレスをやめるとかすればいい。傍受されるされないにかかわらず，コードレスって怖いけどね，いろいろと。特に，パソコンを使っているとノイズとか。大丈夫なんだろうか。微弱型のコードレスホンはノイズを拾ってしまったけど，小電力型だと大丈夫かもしれない。

ところで，盗聴防止コードレスってやつはどうやっているのだろう。小電力型はいくつもチャンネルを持っていて，空きチャンネルを探して使っているわけだが，素早くチャンネルを切り替えて，追いかけられないようにしているのだろうか。それとも，ちゃんとスクランブルかけているのだろうか。

吉田栄作といえば，X68000 XVIのポスターも吉田栄作にX68000を抱えて微笑んでもらったらよかったかもしれない。

無駄話ばかりしていてもしょうがない。「大人のためのX68000」では初めてだが，私にとって，あるいはOh!Xにとっては2回目となる大アンケート結果発表大会を始めよう。わざわざ「大人のためのX68000」に組み込まなくてもいいじゃないか，っていう意見は至極正論だが，正論も負論もこうした事実の前では無力である。

さて，いきなりではあるが，お詫びである。昨年は，Kamikazeの復権も兼ねて，X68000とKamikazeですべてのデータを処理した。誌面に載ったのは写植文字になっていたけど，もとはKamikazeだ。

しかし，荻窪圭が贅沢になったおかげで，今年の表やグラフはすべてMacintoshである。MacintoshIIcx＋Laser Writer NTX-J＋Microsoft Excel 2.2Jである。もちろん，Laser Writerなんてものを私が持っているわけではない。じゃあ，X68000は何をしていたかというと，この原稿を書くのに使っていたのだ。うーん，マルチウィンドウより贅沢なマルチパソコン。

となると，どこが「大人のためのX68000」なんだ？ という意見は至極正論だが，正論も負論も……，どっかで書いた気がする。

X68000ユーザーの割合

昨年度と同じく無作為抽出300枚のアンケートデータをもとに構成した。が，今年は300枚のうち，X68000ユーザーである読者のみを抽出して処理している。X1，MZユーザーの人は悲しいかもしれないが，しかたがないのである。だって，ここは「大人のためのX68000」のページなんだもん（ひでぇ話だ）。

さてさて，300枚中X68000所有者だと宣言しているアンケートは何枚あったか。昨年は300枚中151枚。つまり，約半分であった。

ジャーン。

今年は。なんと，300枚中“244枚”がX68000所有者宣言をしていたのである。

昨年の6割増しだ。どうしたもこうしたもない。全体の80％がX68000持ちだということだ。まあ，X68000も発売以来4年めになるからね。

とりあえず，4年間に出たさまざまな機種ごとの人数を出してある。表1だ。

単一の機種で見ると元祖がいちばん多い。私も元祖だ。4年も前の機械がいちばん多いなんてね。

ではここで，発売年度別に見てみよう。X68000 SUPERの扱いが困るけど，いちおう1990年度に含めておこう。表2である。

これで見ると，1989年度に発売されたマシンを購入したユーザーがいちばん多いことがわかる。元祖とX68000 ACE，X68000 EXPERTの数はあまり変わらないので，X68000 PROを含んでいるだけ多いという感じだ。昨年は出した機種が多いわりに，人数は多くない。

もっとも，元祖X68000ユーザーのOh!X読者の割合と，X68000 SUPERユーザーのOh!X読者の割合が同じとはかぎらないので，そのあたりは考慮に入れなければならない（少なくとも私は元祖ユーザーのほうがOh!X読者の割合が多いと思う。根拠はあんましないけれど）。

さて，来年はここにXVIが加わるわけで，まあ，結果が楽しみである。

X68000を何に使うのか

パソコンの使い道に貴賤なし（これぱっか）。表3を参照していただきたい。昨年度の結果もあわせて記してある。

“ゲーム，プログラミング，音楽，CG”が増えているのに対し，“ワープロ，実務，ビデオの制作”が減っている。

どーゆーことなんだあ！

どうもこうもないわけで，非常にわかりやすい結果である。趣味のパソコンとして

のX68000がさらに推進されただけのことだ。ゲームユーザーが多いのはもう当たり前のこと。というより，ゲームをしないユーザーはかくも少ない。“ゲームもする”ユーザーは，“プログラミング”もする。プログラミングユーザーは来年には80%に届かん，という勢いである。

世の趨勢を見ると，はたしてこの値はOh!X読者全体に通用するのか，はたまたX68000ユーザー全体に通用するのか，事態は予断を許さない。

音楽とCGが増えているのは環境が整ったことを示している。この両者の伸びは顕著だ。その一方でビデオ制作ユーザーが減っているのは，音楽やCGに比べ環境の進歩がなかったことが災いしたといえるだろう。実務やワープロユーザーが減った理由もそのへんである。

X68000の性格がよりはっきりしてきた，来年はどうなるかわかったもんじゃない，そういうことである。

表1　X68000機種別ユーザー数

	人数	割合	機種別割合
元祖	59	24.18%	24.18%
ACE	23	9.43%	22.54%
ACE-HD	32	13.11%	
PRO	28	11.48%	13.52%
PRO-HD	5	2.05%	
EXPERT	33	13.52%	18.44%
EXPERT-HD	12	4.92%	
PRO II	19	7.79%	7.79%
PRO II-HD	0	0.00%	
EXPERT II	11	4.51%	8.20%
EXPERT II-HD	9	3.69%	
SUPER	4	1.64%	5.33%
SUPER-HD	9	3.69%	
全体	244		

表2　年度別ユーザー数

	機種	人数	割合
87年度	**元祖**	59	24.18%
88年度	**ACE**	55	22.54%
89年度	**PRO,EXPERT**	78	31.97%
90年度	**PRO II,EXPERT II,SUPER**	52	21.31%

表3　X68000の使い道

順位	用途	人数	割合	'90年度(参)
1	ゲーム	230	94.26%	90.1%
2	プログラム	193	79.10%	78.8%
3	音楽	129	52.87%	43.0%
4	ワープロ	117	47.95%	56.3%
5	CG	99	40.57%	31.5%
6	通信	47	19.26%	*****
7	入門	43	17.62%	16.6%
8	インテリア	42	17.21%	13.9%
9	実務	30	12.30%	22.5%
10	ビデオの制作	7	2.87%	7.3%
11	周辺機器の制御	6	2.46%	4.0%

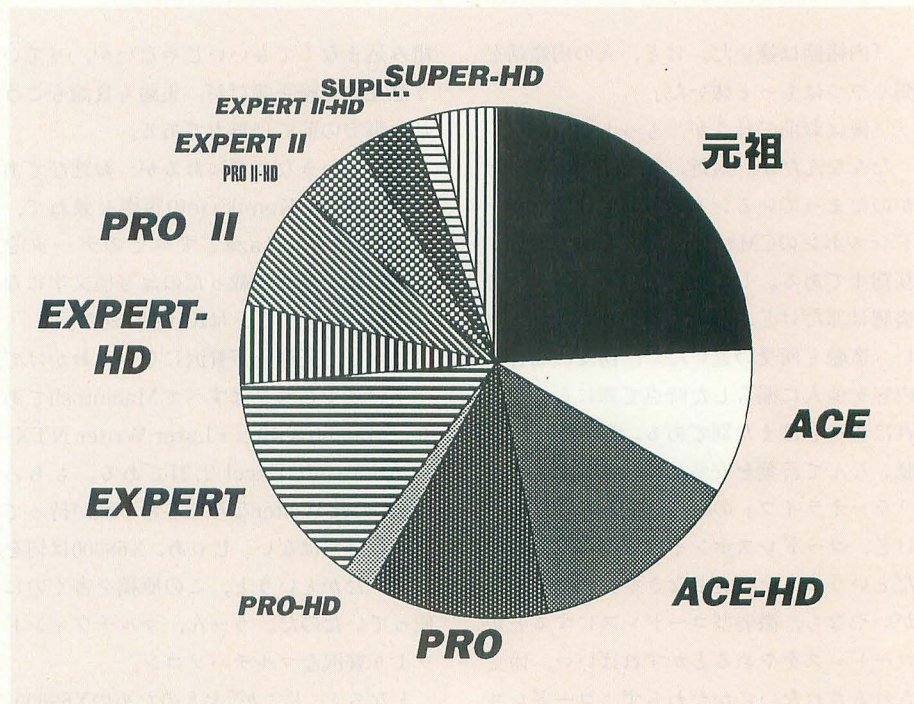
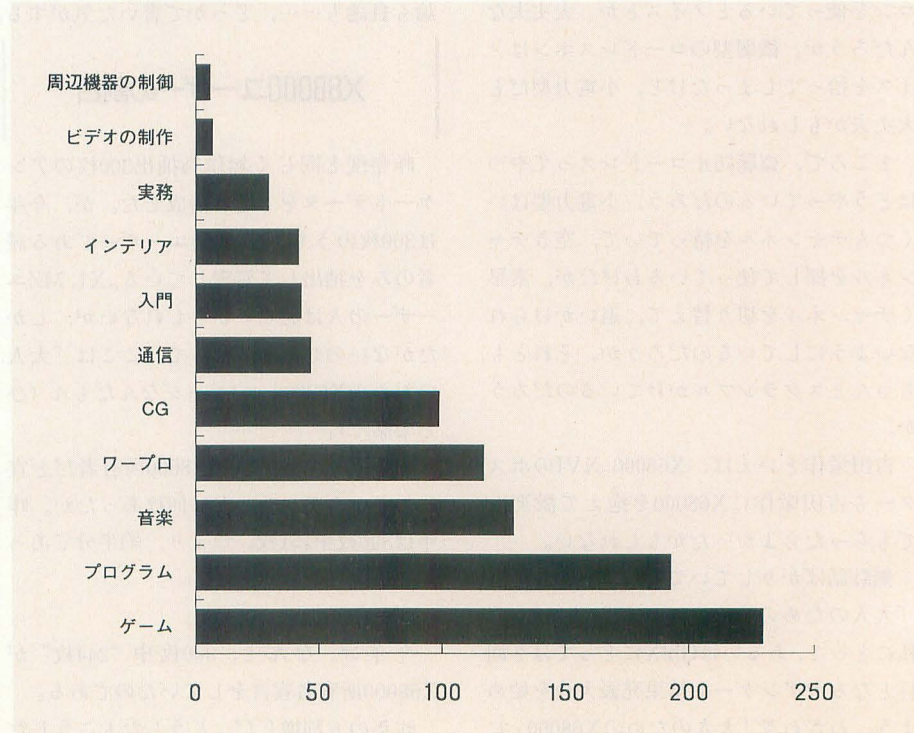
3大言語比べ

なんとまあ，79%のユーザーがプログラミングをする。昨年に比べて増えている。んが，ここが世の不思議なところ。

表4を見よう。各言語の意識調査だ。◎が「開発に使っている」言語，○が「いちおう理解できる」言語，△が「よくわからないが関心のある」言語を示している。

問題はおまけでつけた昨年度データとの比較だ。

なんと，プログラミングをするというユーザーの割合は増えているのに，個々の言語で見ると，見事，減少の傾向を示しているではないか。

グラフ1

グラフ2　X68000の使い道

1)　「よくわからないが関心あり」は増えているので，みんな，謙虚になった
2)　フカシが混じっている
3)　実は，X68000ユーザーはみんなPascalを使っている

さあ，答えは。

どれでもないのである。表をよく見ればわかるとおり，昨年度の数字は合計したら100%を軽く突破してしまうのだ。

つまり，である。昨年と今年では「質問の形態が異なる」のであった。いやあ，焦った焦った。

困ったものである。

質問の違いを差し引いて見ても,「開発に使っている！」と胸を張って答える人の割合は減っているのだけれどね。

それでもまあ，昨年の「アセンブラユーザーのほうがCユーザーより多い！」という状況から，今年は「Cのほうがちょっとばかりメジャーになった」といえるのではないだろうか。昨年,「Cに関心がある」と答えた人が「Cを使える」ようになったのだろう。

第2回ベストライター大賞
──景品も賞状もなしよ──

昨年は執筆量が減っているにもかかわらず，伝統の質実剛健な重みで「祝一平」氏がだんとつでぶっちぎった。今年はどうなったか。祝一平氏の神通力はまだ残っているか。もしかしたら，祝一平という名を知らない読者さえいるかもしれない，今日この頃。

というわけで表5だ。満開製作所にこもった祝一平氏に代わって，昨年3位だった若手のホープ，西川善司氏がトップに躍り出た。昨年の活動量を見ればね。納得。

全体として，(で)氏が泣いているくらいで，昨年とさほど大差はない。丹明彦大先

表4　3大言語の意識調査

		'91 年度	'90年度
BASIC	◎	52.46%	64.90%
	○	37.30%	64.90%
	△	16.39%	10.60%
C	◎	16.80%	21.85%
	○	19.26%	28.48%
	△	54.92%	54.97%
アセンブラ	◎	15.98%	23.18%
	○	14.34%	28.48%
	△	52.05%	43.71%

グラフ3

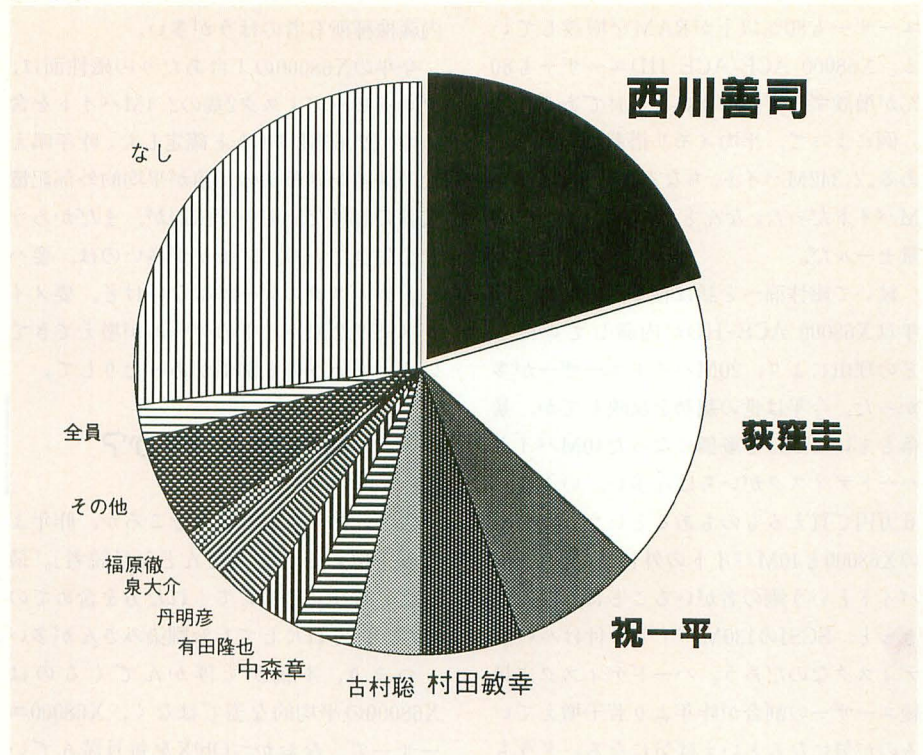



表5　'91年度ベストライター

		票数	割合	対昨年比
1	西川善司	50	20.49%	↑
2	荻窪圭	40	16.39%	→
3	祝一平	18	7.38%	↓
4	村田敏幸	14	5.74%	↑
5	古村聡	10	4.10%	↓
6	中森章	8	3.28%	↑
7	有田隆也	6	2.46%	→
8	丹明彦	6	2.46%	―
9	泉大介	4	1.64%	↓
10	福原徹	3	1.23%	―
	その他	14	5.74%	
	全員	5	2.05%	
	なし	66	27.05%	

生と，福原人間国宝（彼のドット打ち技術は人間国宝に値すると思うぞ）が新しく入ったくらいだ。

昨年とのもっとも大きな違いは，「なし」の多さである。

もともとOh!Xという雑誌は，Oh!MZの時代からそうだったのであるが，祝一平氏，清水和人氏をはじめとする「ライターの個性」が際だった希有のパソコン誌だった。読者は記事の内容と同時に，ライターの個性も楽しんでいたわけだ。

それがX68000の浸透と拡散により，普通の雑誌に近くなった（あるいはノーマルな読者が増えてきた？）のだと思われる。普通，雑誌を読んでいて，ライターの名前まで気にすることはあまりないからね。それでも，Oh!Xから「ライターの個性」が消えることはない，のである。おそらく。

ところで，昨年とあまり変わらないランキングに新風を巻き起こしたい人，いつでも編集部へ。

表6　X68000に搭載されたメモリ

	人数	割合	'90 年度
1MB	32	13.11%	41.06%
2MB	171	70.08%	56.29%
3MB	2	0.82%	
4MB	20	8.20%	1.99%
6MB	17	6.97%	0.66%
7MB	1	0.41%	
??	1		

1台当たりのメモリ
平均　2.342 (MB)

メインメモリ，および磁性面

昨年と同様。今年もやる。

まず，メインメモリ搭載量を示したのが表6だ。見てのとおり，昨年と比べ大容量メモリユーザーが増えている。とても増えている。昨年，2Mバイト以上のユーザーは60%弱だったのに対し，今年は80%を越えた。ほとんどのユーザーが増設，あるいははじめから2Mバイトを搭載したマシンを購入しているわけだ。いまのX68000の環境を考えれば，当然の結果といえよう。ち

なみに表には出ていないけれども，元祖のユーザーも80%以上がRAMを増設している。X68000 ACE/ACE-HDユーザーも80%が増設ずみだ。そういうわけである。

例によって，平均メモリ搭載量も出してある。2.342Mバイト。ちなみに昨年は1.656Mバイトだった。なんと，686Kバイトの増量セールだ。

続いて磁性面へと話は移る。表7だ。昨年はX68000 ACE-HDが内蔵していたなどの理由により，20Mバイトユーザーが多かった。今年は世の趨勢を反映してか，暴落ともいえるほど廉価になった40Mバイトハードディスクがいちばん多い。いまや，6万円で買えるものもあるという。我が家のX68000も40Mバイトの外付けだ。130Mバイトという剛の者がいることにも注意。きっと，SCSIの130Mバイト外付けハードディスクなのだろう。ハードディスク未接続ユーザーの割合が昨年より若干増えているのが気になるといえば気になる。どうもX68000 PROユーザーに多いようだ。どうしてもX68000がほしいけれど，お金がぎりぎりしかなかった，というユーザーが多いのではないだろうか。ちなみに，外付けハードディスク所有者より，ハードディスク内蔵機種所有者のほうが多い。

今年のX68000の1台あたりの磁性面は，フロッピーディスク2基の2.4Mバイトを含めて，22.65Mバイトと確定した。昨年唱えた，「メインメモリの10倍が平均的外部記憶装置の容量だ」という法則が，まだかろうじて生きている。メモリが多いのは，要ハードディスクのゲームはないけど，要メインメモリ2Mバイトのゲームが増えてきている，というのと関係があったりして。

表7　X68000の所有する磁性面

	人数	割合	'90年度
20 MB	36	14.75%	26.49%
40 MB	52	21.31%	19.21%
60 MB	1	0.41%	
80 MB	18	7.38%	1.32%
120 MB	1	0.41%	
130 MB	4	1.64%	
0	132	0.54098	0.5298

1台あたりの磁性面(FD2.4MB含む)

平均　22.65 (MB)

表8　Oh!Xを……

Oh!Xを	人数	割合
初めて買った	2	0.85%
ときどき買う	6	2.54%
最近よく買う	29	12.29%
ほとんど毎月	199	84.32%

表9
X68000ユーザーの所有するハード&ソフト

	人数	所有率
SX-Window	174	71.31%
C	130	53.28%
Z's staff PRO68K	79	32.38%
プリンタ	147	60.25%
MIDI音源	69	28.28%

毎月読んでますか？

表8。昨年に引き続きどころか，昨年より若干増えている「ほとんど毎月読者」。「最近よく買う」と答えてくれた方を含めての話だが，それにしてもお馴染みさんが多い。

つまり，本稿から浮かんでくるのはX68000の平均的な姿ではなく，X68000ユーザーで，なおかつOh!Xを毎月読んでいて，なおかつこういうアンケートにちゃんと答えてくれた読者の姿なのである。この域に達しているユーザーは，世間ではパワーユーザーの部類に入ってしまうかもしれない。

初めて買った人も，御用とお急ぎでなかったら，アンケート用紙に思いのたけをぶつけよう。

代表的なハード&ソフト

今回はちょいと規模を縮小して，ソフトはSX-WINDOW，XC，Z'sSTAFF PRO-68Kの3本，ハードはプリンタとMIDI音源についてだけ数えてみた（表9）。

意外だったのは，SX-WINDOW所有者が結構多いこと。ハードディスク所有者より多いのが面白い。安かったから，ゲームでも買うつもりで遊んでみたのだろう。バージョンアップもされることだし，気に入った人はハードディスクを買うべきだな。

XC所有者も半分以上。昨年は49%だったのが今年は53%だ。本アンケート回答者がX68000ユーザー全体に比べて，パワーユーザー指向だったとしても，半分がXCを持っているというのはなかなかだ。

そのわりに，プリンタの所有率が下がっていて面白いけどね。

今回最高のポイントがMIDI音源だ。昨年

表10　Oh!X読者所有機種の推移

		Jun-86	Apr-87	Apr-88	Mar-89	Mar-90	Apr-91
MZ	80K/C/1200	20	18	27	16	13	17
	700/1500	196	155	118	84	66	67
	80B/2000/2200	220	79	70	64	42	38
	2500	0	145	113	53	52	23
X1	マニアタイプ	146	91	73	56	27	23
	C/Cs/Ck	173	94	65	72	46	37
	D	54	32	31	25	16	8
	F/G	0	105	105	83	77	53
	twin	0	0	13	1	7	4
	turbo/II/III	140	326	333	274	143	161
	turboZ/II/III	0	39	112	129	114	89
X68000	初代	0	0	269	220	152	162
	ACE/HD	0	0	0	209	215	186
	PRO/HD	0	0	0	0	133	167
	PRO II/HD	0	0	0	0	0	51
	EXPERT/HD	0	0	0	0	144	166
	EXPERT II/HD	0	0	0	0	0	81
	SUPER/HD	0	0	0	0	0	45
NEC PC		5	34	60	57	25	55
FM		2	16	13	5	5	8
ポケコン		6	31	25	55	43	22
その他		18	27	47	71	25	75
なし		7	20	19	19	22	19

は10%に満たなかったMIDI音源所有者が，今年は一挙に3倍の28%が所有しているのだ。そっと内訳を教えると，いちばん多いのがMT-32，次いでCM-64だ。この2機種（といってもどちらも同じようなものだが）で4割以上だ。あとは混戦状態だが，かろうじて，あのばかでかいM1が3位につけている。

MIDI隆盛もわかるね。

愛用機種推移

いよいよクライマックスである。というわけで表10を見てくれ。

昨年のデータに1991年分のデータを付け加えたものだ。これは先月号に載っていた愛読者ハガキ1000枚無作為抽出分のデータである。こういう結果だ。X68000などは機種ごとに分かれているので全体像はつかみにくいかもしれない。

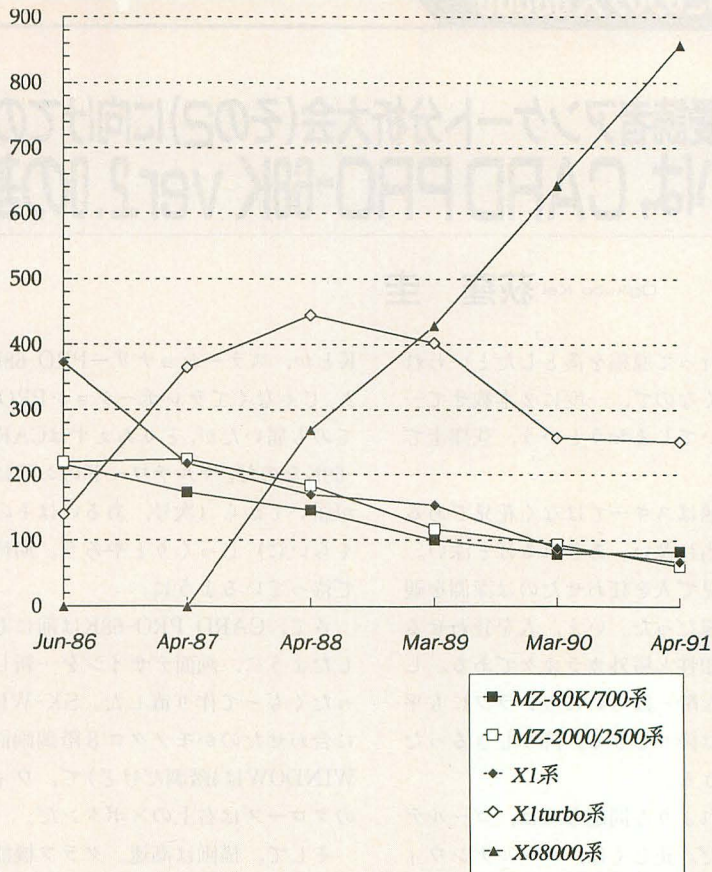
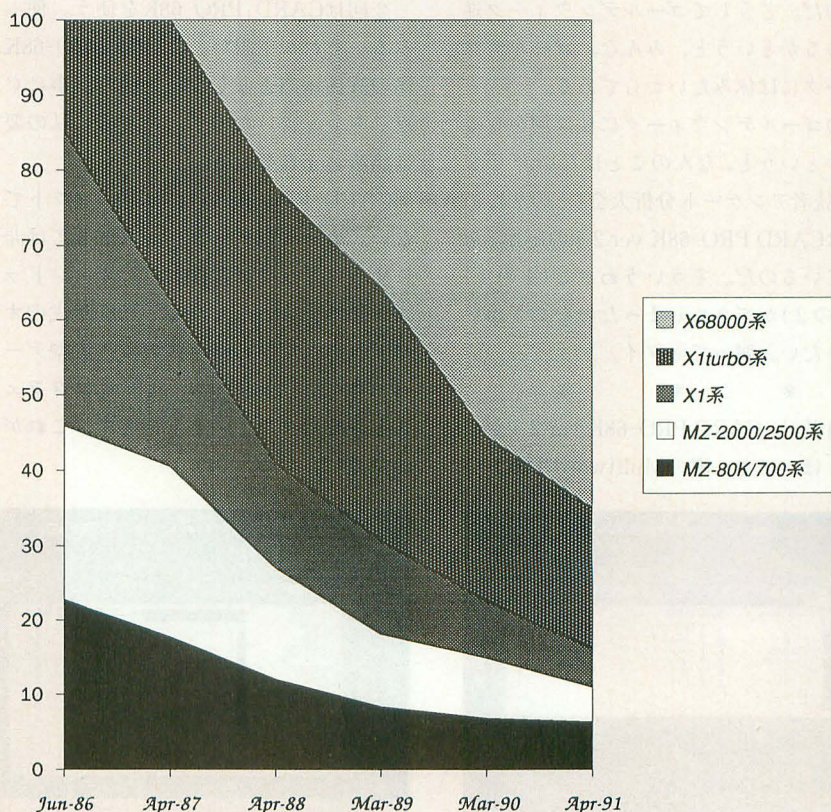
そこで，シリーズごとにまとめてグラフ化したのがグラフ4と5だ。グラフ5は各機種の割合が示してある。「X68000勢力の伸び」の図が見えてくる。面白いのは，昨年から今年にかけて，8ビット機があまり減っていないこと。MZ-2000/2500とX1は勢力を落としているが，MZ-80K/700やX1turboは結構がんばっている。いや，がんばっているのはMZ-80K/700やX1turboではなく，そのユーザーなんだけどね。いいパソコンはいいパソコンなので，ずっと使い続けていただきたいと思う。

順番が前後したが，グラフ4を見てほしい。こういう結果である。特にX68000なんてのは，昨年私が予言したとおりの結果だ。来年はどうなるか楽しみである。

次回予告

X68000ユーザーは増えていて，なおかつ，みんな精力的に使っているということがわかった。パソコンの使い道の多様化現象も見えてきた。非常に面白い。

さて，次回は「第2回　愛読者アンケート分析大会（その2）に向けての準備あるいは，CARD PRO-68K ver.2.0の基礎」の予定である。予定は未定だが，実現されるだろう。次回が載るのは1991年6月号なので，お間違いなきよう。

グラフ4　Oh!X読者の所有機種推移その1（1000人中の所有者数）

グラフ5　Oh!X読者所有機種推移その2（1000人中1538台の内訳）

第2回 愛読者アンケート分析大会(その2)に向けての準備 あるいは,CARD PRO-68K ver.2.0の基礎

Ogikubo Kei 荻窪 圭

「大人のためのX68000」はもう読んだ気がするという人，気のせいではありません。今月の「大人のためのX68000」はもう1回。来月行う予定のアンケート分析（その2）を例に,「CARD PRO-68K ver.2.0」の紹介をしていきます。

スキーに行って原稿を落としたといわれるのもしゃくなので，一度に2本載せて一気に追いついてしまおうという，荻窪圭である。

さて，問題はスキーではなく花見である。夜の桜は本当に深い。あきれるほど深い。かつて，花見で人を狂わせたのは深淵を覗かせる桜の嵐だった。いま，人を狂わせるのはスケベ根性と屋外カラオケである。しかし，どんな酔っぱらいにもシラフにも平等に花びらは降りしきる。降りしきるったら，降りしきる。

さらにそれよりも問題なのは，ゴールデンウィークだ。正しくは，ゴールデンウィーク進行という。締め切りがいつもよりずっと早いというのを，私はすっかり忘れていたのだ。どうしてゴールデンウィーク進行があるかというと，みんな，ゴールデンウィークには休みたいからである。

そのゴールデンウィークに私は何をしているかというと，なんのことはない,「第2回 愛読者アンケート分析大会(その2),あるいはCARD PRO-68K ver.2.0の応用」を書いているのだ。そういうわけで(その1)と(その2)ができてしまったのをご了承いただきたい。腐ってもタイ。

＊　＊　＊

某月某日。CARD PRO-68K ver.2.0が届いた。ほかにも，噂のMultiword PRO-68Kとか，ステーショナリーPRO-68Kのver.2，じゃなくてテレポーションPRO-68Kってのも届いたが，とりあえずはCARD PRO-68Kなのだ。マルチワードのお話は完成品が届いてから（次号，あるいはその次の号くらいに）じっくりとやろう。期待しないで待っているように。

さて，CARD PRO-68Kは前にもお伝えしたように，画面デザインを一新した。まったくもって作り直した。SX-WINDOWに合わせたのかモノクロ8階調画面（SX-WINDOWは4階調だけど)で，ウィンドウのクローズは右上の×ボタンだ。

そして，描画は高速。グラフ機能付加。キーボードマクロもあるし，一覧表入力もできた。いたれりつくせり。

今回はCARD PRO-68Kを使う。使ってみる。それが主眼だ。CARD PRO-68K自体の評価はあまりしない。囲み記事にぐちゃぐちゃと書いたので，そちらに私の要望は詰め込まれている。

CARD PRO-68K自体は悪いソフトではない。カード型データベースとしては非常にオーソドックスな作りだ。オーソドックスすぎるくらい。しかも，1980年代のオーソドックスだ。1980年代のカード型データベースに，1990年のユーザーインタフェイスを乗せたソフト。とりあえず，これが私の意見だ。

データベースの設計

CARD PRO-68Kにデータを入力する。方法は2つある。他の形式でファイルを作っておき，それをコンバートする方法と，1つひとつ手で入力していく方法だ。

物事の順序からして，1つひとつ手で入力するほうを選ぼう。

メニューバーにずらっと並んだ機能。このメニューバーも移動できるのだが，あまりメリットはないからどうでもいい。

メニューバーのいちばん左の“データベース”ってのがポイントだ。

この“データベース”のところをクリックすると，べろんとプルダウンメニューが出る。そこで“定義作成・変更”を選択する。新しくデータベースを作るときには，その定義をしなければならんのだ。

こいつがまた面倒。渡る世間の商売人が仕事で使うなら，自分の使おうとするデータベースの姿がすぐに思い浮かぶだろう。しかし，たいていの場合はそうではない。が，必要なものはしょうがないので，定義する。

“定義の作成・変更”を選ぶと，ファイル選択のダイアログが現れる。変更なら変更したいデータベースを選択し，新規作成なら“新規ファイル”をクリックする。

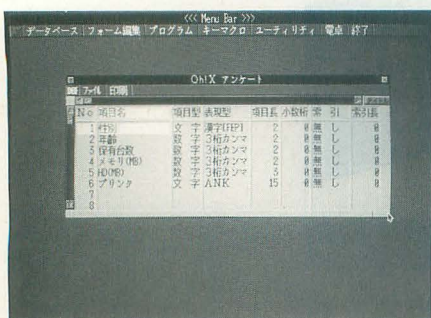



写真1　データベース定義の図

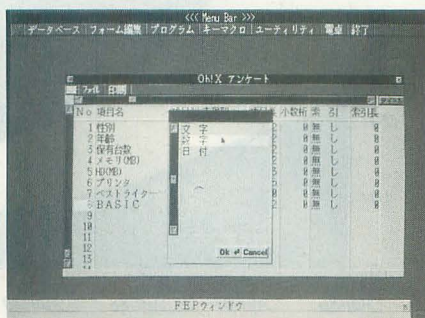



写真2　項目型を選ぶの図

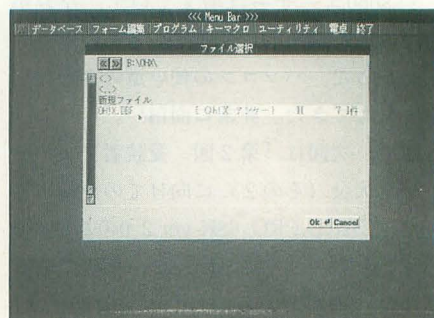



写真3　まず対象ファイル選択からはじまる

すると，データベース定義専用ウィンドウが開く。このとき，定義ウィンドウの中，タイトルバーの下にメニューバーが現れる。まあ，このあたりは画面写真を見てほしい。CARD PRO-68Kの基本だから，興味ある人はチェックしておくように。タイトルバーの名前は任意だ。何でもいい。あとから変更したいときは，“タイトル”ボタンを押せばいい。

さて，画面のいちばん上にある“Menu Bar”。これがメインのメニューバーだ。こいつはダイアログが開いているとき以外なら，いつでも使うことができる。そのときに開いているウィンドウには無関係に動作する。

開いているウィンドウの操作をしたいときは，ウィンドウのメニューバーを使う。これでそのときに必要な処理はすべてまかなえるはずだ。

定義ウィンドウの場合は，ファイルと印刷しかないから追求することもない。

この定義ウィンドウでデータベースの設計を行う。各項目に設定できるのは，写真1のとおりだ。項目型や表現型は選択式である。

項目型は“文字/数字/日付”の3つから。表現型は，項目が数字型なら“標準/日本語/3桁カンマ/4桁カンマ”の4つから，文字型なら“日本語[FEP]/ANK”の2つから選ぶようになっている。

データベース設計の入力というのはとかく，面倒でいらいらして気に入らないものだったが，マウスオペレーションのおかげで，比較的楽にこなすことが可能だ。

今回は「せっかく集まったアンケート結果なのに，集計だけ取って終わりにしては，ばちがあたる」という心がけにより，300枚ほど入力して，いろいろと加工してみる予定だ。

写真4　一覧表画面で編集メニューをプルダウン

何をするかというと，膨大なデータから，未成年と成年ではどちらが多くメモリを積んでいるか，とか，使用言語別に，いちばん多い年齢層を出してみるとか，レイトレーシングしているユーザーの何割がコプロを搭載しているかとか，そういうことだ。搭載メモリと接続ハードディスク容量に相関関係はあるか，というようなことも。それには，一度データベースにデータを蓄えてから，抽出したり，集計したりするのがコンピュータっぽい。

さて，データベースの定義が終わったら，データ入力である。

データの入力

メニューバーのデータベースから，“操作”を選択する。するってえと，データベース編集ウィンドウが開く。こいつがすべての基本画面となるのだ。

タイトルは“Oh!X アンケート”。

このウィンドウでは何をするか。

まず，そのデータベースに対してどういう操作をするか決める。メニューバーにある“編集”をクリックすると，写真4のようにメニューがいろいろと現れる。代表的なのが，参照，追加，変更，削除だ。参照にしておくと，見るだけで書けないので，つまらない。データ入力するときは“追加”にする。

続いて，どういう画面でデータ操作をするか。それが編集の右の右にある“画面”ってやつだ。ここでは，標準画面，自由画面，一覧表画面，グラフ画面の4つから使いたい画面を選ぶ。使う画面を選ぶだけであって，画面を設計するのはまた別の場所なのだ。

もちろん，まだ自分で設計した画面はない。ここでは標準画面か，一覧表画面の標準リスト画面を選択することになる。

特に1レコード（つまりデータ1件あたりの項目数や桁数）が多すぎなければ，前後を参照しながら入力できる，標準リスト画面が最適である。カット&ペーストもできて便利だ。

さて，データベースを作るパターンは，だいたい4つのケースに分けることが可能である。

いまどきのデータベース

CARD PRO-68Kは確実に前のバージョンよりよくなった。機能が増えたということより，処理の高速化や，気楽に扱えるようになったことのほうが大きい。

しかし，前にも書いたが，X68000のユーザー層を考えた場合(昔の記事を参照のこと)，あまりにも普通のカード型データベースすぎる。一世代前のスペックではないか，というところも少なからずある。

たとえば，いまどきのパーソナルユース向けデータベースは，少なくとも「項目設定をせずにスプレッドシート感覚で初期入力が可能である」ことと，「可変長文字列項目」を持っていることが条件だといえよう。前者に関してはサポートしていないデータベースソフトも多いが，後者は必須だ。

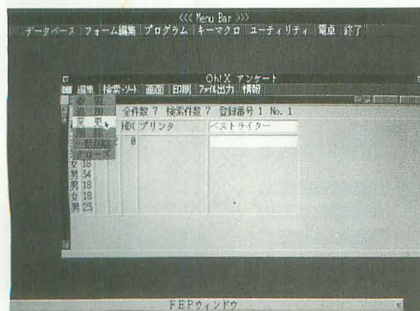
次いで，項目型。なんと，CARD PRO-68Kは，選択型項目がないのである。選択型というのは，あらかじめ設定しておいた選択肢の中から選んで埋め込むもの。つまり，選択肢からマウスでクリックするなり，カーソルキーで選ぶなりできるような項目型だ。三者択一の項目，なんてのは日常茶飯事だし，そうでなくても，“男/女”なんてのを毎回変換しながら打ち込むのはたわけたことだ。さらに，項目型は多ければ多いほどいい。年齢型（生年月日を入れておくと，自動的に計算される）とか，郵便番号型とか，電話番号型とか，人名型（？）とか，いろいろあるとユーザーが楽だ。

個人的には，インデックスなんてなくてもいいと思っている。あれは，大型コンピュータででかいデータを大量に扱っていたときの残滓だ。パソコンのデータベースは大型機のデータベース概念にパソコンのインタフェイスを載せたものではなく，根本的にパソコンのためにあるものなのだ。

そして，集計機能。BASICライクなプログラムを使えば，CARD PRO-68Kはたいていのことはできてしまうようだが，ちょっとした集計くらいはもっと簡単に実現できてほしい。

文句をつけるのはあまりにも安易なのでやりたくはないのだが，次のバージョンでは（おそらく，SX-WINDOW対応版となるだろうが），なんとか，1990年代のパーソナルデータベースを目指してもらいたいと思う。

いうだけいって去っていくのも気がひけるが，X68000の性格を考えると，ビジネスソフトとしてのデータベースソフトもいいが，パーソナルユーザーのためのデータベースソフトも必要なのだ。

究極のデータベース「テキストファイル」。究極のデータベースマネジメントシステム，エディタとフィルタコマンド。なんていう時代は終わろうとしているしね。

ひとつは，なんらかの形で（紙でも何でもいいから）たまったデータがある程度あって，パソコンにぶち込みたいとき。住所録なんかはそうだし,蔵書管理ってのも(そういうことをする人がいればの話だが）そうだ。こういう場合は，まずたまったものをエディタで打ち込んで，どさっとコンバートをかけるのがいい。

別の形式のファイルや，旧バージョンのデータベースを持ってきたい場合ももちろん存在する。CARD PRO-68K ver.2.0は結構，しっかりしたデータコンバート機能を持っている。テキストファイルに落ちていればたいていの形式が扱えるのだ。

続いて，今回のアンケート分析のように，使用するデータが決まっていて，一度打ち込んでしまえばあとは検索・加工するだけの場合。これもエディタ向きだ。

最後は，はじめから少しずつ入力していく場合。伝票管理なんかがそうだ。こういうときだけ，はじめから帳票設計をしてやるべきだろう。

データベースの画面設計

というわけで，画面設計である。通常，こういった設計は画面表示に留まるものではないため，大人の世界にしかない“帳票”という言葉を使う。

たとえば，入力画面を作る。お仕事ではあらかじめ決まった形式の伝票やら請求書やらがあるので，それに合わせて作ると，事務の女の子が入力しやすい。私には関係ないな。

入力しやすいようにコメントを入れた画面を作っておくと，いろいろと便利。

そういうときはわざわざ設計をする必要がない。ちゃんと標準画面が用意されているからだ。

CARD PRO-68Kはマウス１個でいろいろと操作できるのでよさそうだが，実は，慣れるまで結構面倒だったりする。ユーザーが文字列と項目と罫線の３種類のデータの違いさえチェックしておけばいいのだけれど，どれも違いなく使えるのが，目指すべきところだ。

さて，データ入力時は設計するまでもない画面でも，データを参照するときにはなかなか便利だ。たとえば，１件に対して項目が20個も30個もあった場合など。データベースのデータを検索するときは，そのときどきで参照したい項目は決まっているのだから，いくつかのパターンに分け，その都度必要な項目だけを参照できるようにする，なんてのは非常にありがちで効果的なものだ。

同じことは一覧表画面でもいえる。この一覧表画面も，どの項目を出力するかを設定できるのだ。そうしないと，レコードが長いからといって横スクロールするのはぞっとしないしね。

肝心の使い勝手だが，そういいものではない。まあ，カーソルキーとファンクションキーで同じような作業をさせるDOSマシンよりは善良だが，左と上にあるスクロールバーとか，自由設計画面の印刷ができないといった欠点をさらけだしている。

日本に流通しているデータベースソフトの中ではそう悪い出来ではないが，X68000は某DOSマシンのようにソフトの数で勝負できないので，１つひとつの質を（特に，シャープブランドで出すかぎり）高めなければどうしようもないのだ。

おまけ

検索，ソート，グラフ，プログラムなど，データベースの醍醐味は次回の(その２)へ持ち越しである。それまでにデータを用意しておくから待っているように。

ところで，CARD PRO-68Kのver.2.0にはユニークな機能がついた。全体としてビジネス色が強いデータベースソフトなのだが，こういうところがX68000だ。というのは，背景を自分で設定できるのである。

背景の作り方は簡単だ。コマンドシェルの上において，自分の用意したグラフィックをCARD PROで扱えるモノクロデータに変換するプログラムを使って，背景用のデータを作る。あとは，CARD PRO-68K上のユーティリティメニューの環境設定で背景ファイルを設定するだけ。

SX-WINDOWにもない，この背景設定機能。モノクログラフィックを背景に操作するのはなかなか心がなごんでよろしい。

＊　＊　＊

次号予告。今回のデータベースをきちんと完成し，プログラム機能を使ってさらなる情報の深淵へと突入する。

荻窪圭にゴールデンウィークはない。きっとない。ないのではないかと思う。でも，無理やり遊んでしまうかもしれない。

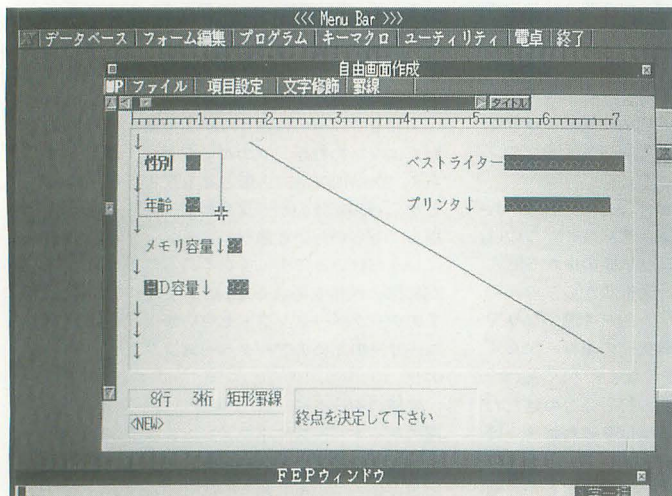



写真5　画面設計をしてみる

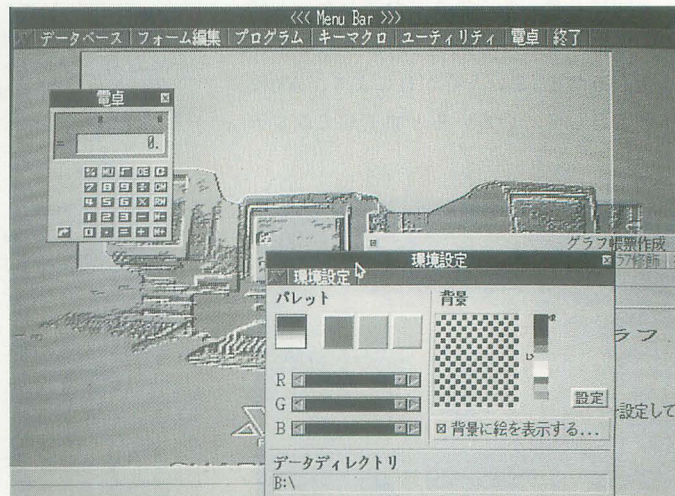



写真6　背景に絵を置くことも可能になった

X68000CARDDRV用カードゲーム

ChristmasTree

Ohkubo Akihiro 大久保 明弘

しばらくぶりのカードゲームです。今回はカードのレイアウトが美しい「Christmas Tree」，ひとり遊び用のトランプゲームです。あまり成功率が高くないので，恋占いなどには向いていないかもしれませんね。

Klondikeのようで Klondikeでない，TENのようでTENでない，それはなにかとたずねたら，そうです。ChristmasTreeなんです。なぜこのような名前なのかは写真を見ればわかりますね。最初，カードはクリスマスツリーのように配置されます。

このゲームはひとり占い用のトランプゲームをもとにしています。場に並べられたカードがすべて取り除けたなら，かなり強運の持ち主だといえるでしょう。

簡単にルールを説明しましょう。使用するカードはジョーカーを除いた52枚。これを図のように並べます。手札のうちの1枚を捨て札の台とし，場札のうち「上に乗っているカードがなにもないもの」と比べます。もし，スート（種類）や数字の同じカードが場にあったら場から捨て，今度はそのカードを台札としてゲームを繰り返します。どうしても進まないときは手札を出して新しい台札にします。こうして，場にあるすべてのカードを捨てればゲームクリア，手札がなくなったらゲームオーバーになります。

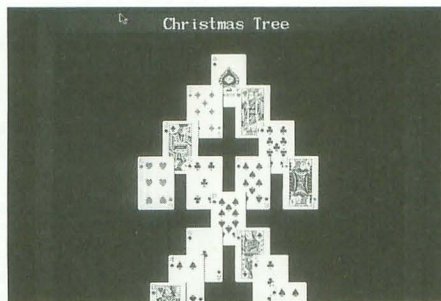
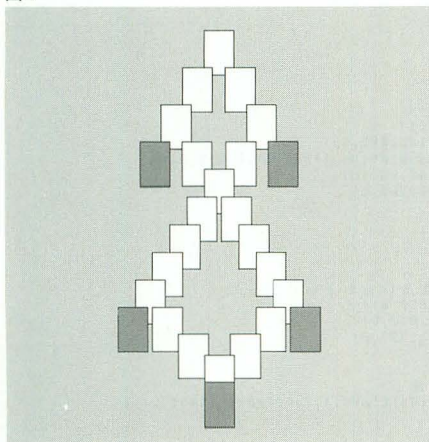
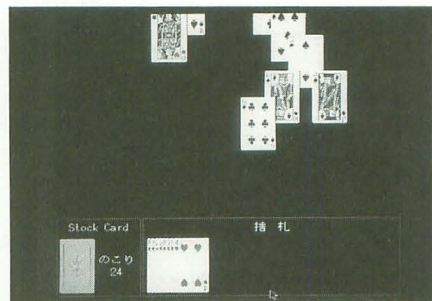



図1

プログラムについて

このゲームを作るうえでいちばんやっかいだった点は「どのカードを選択したか？」でした。レイアウトを見るとかなり不規則にカードが並んでいます。これではマウスがクリックされたXY座標でカードを判定するのは難しいものがあります。一度はここで挫折してChristmasTree制作は一時中断になりました。しかし，その5時間後，いい方法がひらめきました。「仮想画面を使えばよいのでは？」

画面の右半分が余っているので，そこを仮想画面とし，カードを塗り分けてマウスがクリックされた座標の色を調べることによって，どのカードが選択されたかを判断しました。

たとえば，画面上にc_put (X,Y,1) でスペードのAを表示します。そして仮想画面には fill (X+512, Y, X+47+512, Y+95,1) のように1の色でフィルボックスを描きます。そして，スペードのAの位置でマウスがクリックされたなら，if point (MX+512, MY)=1 で判断すればよいわけです。

ところが，まだ落とし穴がありました。場に出すカードは全部で26枚あります。ところがカードデータは16色対応になっており，256色は使えません。そこで奥義「境界線」を使いました。詳しくはプログラムを解析してみてください（ブレイクしてhome (0, 512, 0) などを実行するとよい）。

次の問題点は「選択したカードが取れるか？」でした。いちばん上に出ているカードしか捨てることができませんから，上にカードが乗っているかどうかをチェックしなければなりません。場に出ているカードに順に番号をつけたとき（0～25），0のカードが取り除けるためには1と2のカードが取り除かれなければなりません。そこで，場のカード1つひとつにデータを用意しました。プログラムでは dim int chk (52)=……でデータを用意しています。

リスト1

```
 10 /*
 20 /* Christmas Tree
 30 /*    Written by Azuron 4.6(Sat.)
 40 /*
 50 int i,sute,cp,fin
 60 int pasa=1,boo=2
 70 dim int xy(52)={ /* カードのXY座標
 80     +232,  0,200, 56,264, 56,168,120,296,120,
 90     +136,184,200,184,264,184,328,184,232,248,
100     +200,312,264,312,176,360,288,360,152,408,
110     +312,408,128,456,336,456, 96,520,160,520,
120     +304,520,368,520,192,568,272,568,232,600,
130     +232,664}
140 dim int chk(52)={ /* 間違えないでゆっくり入力
150     + 1, 2, 3,26, 4,26, 5, 6, 7, 8,
160     +26,26, 9,26, 9,26,26,26,10,11,
170     +12,26,13,26,14,26,15,26,16,26,
180     +17,26,18,19,20,21,26,26,22,26,
190     +23,26,26,26,24,26,24,26,25,26,
200     +26,26}
210 dim int cd(52),cflg(27)
220 /*
230 prep()
240 music()
250 /* メイン
260 repeat
270   vinit()
280   shuffle()
```

```
 290   layout()
 300   stock()
 310   suteru(0,-1)
 320   repeat
 330     fin=choice()
 340   until fin<>0
 350   if fin=2 then perfect()
 360   fin=replay(fin)
 370   again()
 380 until fin=0
 390 /* 終了
 400 screen 2,0,1,1
 410 mouse(0)
 420 end
 430 /*
 440 func choice() /* カードを選ぶ
 450   int n,bl,br,mx,my
 460   repeat
 470     msstat(n,n,bl,br)
 480     mspos(mx,my)
 490     if my<513 then home(0,0,my)
 500   until bl+br<>0
 510   if mx<80 and my>445 then {
 520     if suteru(0,0)=1 then return(1)
 530   }
 540   if my>34 and my<414 then {
 550     if toru(mx,my*2)=1 then return(2)
 560   }
 570   return(0)
 580 endfunc
 590 /*
 600 func toru(x,y) /* カードを取る
 610   int co,c,d,xx,yy,mx,my
 620   co=point(x+512,y)
 630   if co=0 then oto(boo):return(0)
 640   if y>=478 and co<13 then co=co+14 /* y=478 dead line
 650   co=co-1
 660   c=cd(co):d=cd(cp)
 670   if check(co,c,d)=1 then oto(boo):return(0)
 680   mspos(mx,my)
 690   msarea(0,my,511,my+1)
 700   xx=xy(co*2):yy=xy(co*2+1)
 710   fill(xx,yy+69,xx+48,yy+165,8)
 720   cflg(co)=0
 730   suteru(1,c)
 740   layout()
 750   if cflg(0)=0 then return(1)
 760   return(0)
 770 endfunc
 780 /*
 790 func layout() /* カードを場に並べる
 800   int i,x,y,c=0
 810   mouse(2)
 820   fill(608,70,928,830,0)
 830   for i=0 to 25
 840     if i=14 then c=1 else c=c+1
 850     if cflg(i)=0 then continue
 860     x=xy(i*2):y=xy(i*2+1)+70
 870     c_put(x,y,cd(i))
 880     fill(x+512,y,x+559,y+95,c)
 890   next
 900   msarea(0,0,511,511)
 910   mouse(1)
 920 endfunc
 930 /*
 940 func suteru(sw,cc) /* カードを捨てる
 950   int x=146,y=890,xx
 960   if cp=51 then return(1)
 970   xx=sute*5+x
 980   if sw=0 then {
 990     if cc=-1 then {
1000       c_put(x,y,cd(cp))
1010     } else {
1020       c_put(xx,y,cd(cp+1))
1030       cp=cp+1
1040     }
1050     oto(pasa)
1060     stock()
1070   }
1080   if sw=1 then {
1090     c_put(xx,y,cc)
1100     oto(pasa)
1110     cd(cp)=cc
1120   }
1130   sute=sute+1
1140   return(0)
1150 endfunc
1160 /*
1170 func stock() /* ストックカード表示
1180   int x
1190   x=114-len(itoa(51-cp))*8
1200   fill(98,940,115,956,8)
1210   symbol(x,940,itoa(51-cp),1,1,1,15,0)
1220 endfunc
1230 /*
1240 func shuffle() /* シャッフル
1250   int i,a,b,k
1260   for i=1 to 37
1270     a=int(rnd()*52):b=int(rnd()*52)
1280     k=cd(a):cd(a)=cd(b):cd(b)=k
1290   next
1300 endfunc
1310 /*
1320 func vinit() /* 変数初期化
1330   for i=0 to 51:cd(i)=i+1:next
1340   for i=0 to 25:cflg(i)=1:next
1350   cflg(26)=0
1360   sute=0:cp=26
1370 endfunc
1380 /*
1390 func suit(c) /* 同じスートか?
1400   return((c-1)¥13)
1410 endfunc
1420 /*
1430 func same(c) /* 同じ数か?
1440   c=c-1
1450   return(c-(c¥13)*13))
1460 endfunc
1470 /*
1480 func music() /* 音楽
1490   for i=1 to 3
1500     m_alloc(i,500):m_assign(i,i)
1510   next
1520   m_tempo(200)
1530   m_trk(1,"@59v15c8")
1540   m_trk(2,"q8@15v13o3c2")
1550   m_trk(3,"r2")
1560 endfunc
1570 /*
1580 func oto(t) /* 効果音
1590   m_play(t)
1600   repeat:until m_stat(t)=0
1610 endfunc
1620 /*
1630 func wait(t) /* ウエイト
1640   m_play(t+2)
1650   repeat:until m_stat(t+2)=0
1660 endfunc
1670 /*
1680 func check(p,c,d) /* カードが取れるかのチェック
1690   p=p*2
1700   if cflg(chk(p))=1 or cflg(chk(p+1))=1 then return(1)
1710   if suit(c)<>suit(d) and same(c)<>same(d) then return(1)
1720   return(0)
1730 endfunc
1740 /*
1750 func replay(sw) /* リプレイ?
1760   int n,bl,br,mx,my,y
1770   wait(1)
1780   home(0,0,0)
1790   if sw=2 then y=340 else y=200
1800   fill(146,y,386,y+80,12)
1810   box(146,y,386,y+80,15)
1820   box(148,y+2,384,y+78,15)
1830   box(200,y+48,236,y+67,5)
1840   fill(712,y+48,748,y+67,1)
1850   box(290,y+48,326,y+67,5)
1860   fill(802,y+48,838,y+67,2)
1870   symbol(210,y+14,"Ｒｅｐｌａｙ?",1,1,1,3,0)
1880   symbol(206,y+50,"Yes",1,1,1,0,0)
1890   symbol(301,y+50,"No",1,1,1,0,0)
1900   msarea(200,y+48,326,y+67)
1910   repeat
1920     msstat(n,n,bl,br)
1930     mspos(mx,my)
1940   until bl+br<>0
1950   msarea(0,0,511,511)
1960   if point(mx+512,my)=1 then return(1)
1970   return(0)
1980 endfunc
1990 /*
2000 func perfect() /* パーフェクト!!
2010   int i,x=32
2020   dim int c(576)
2030   str pe[18]="Ｐｅｒｆｅｃｔ!!"
2040   home(0,0,0)
2050   for i=1 to 9
2060     c_put(x,200,0)
2070     x=x+50
2080   next
2090   wait(1)
2100   x=32
2110   for i=0 to 8
2120     c_put(100,600,1)
2130     fill(100,602,145,692,15)
2140     symbol(100,624,mid$(pe,i*2+1,2),2,2,2,5,0)
2150     get(100,600,147,695,c)
2160     put(x,200,x+47,295,c)
2170     oto(pasa)
2180     x=x+50
2190   next
2200 endfunc
2210 /*
2220 func again() /* もういっちょいく?
2230   fill(24,70,487,830,8)
2240   fill(608,70,928,830,0)
2250   fill(142,880,482,987,8)
2260 endfunc
2270 /*
2280 func prep() /* 準備
2290   randomize(val(mid$(time$,4,2)+right$(time$,2)))
2300   screen 1,0,1,1
2310   console ,,0
2320   window(0,0,1023,1023)
2330   palet(8,rgb(0,6,0))
2340   palet(2,rgb(9,6,9))
2350   fill(0,0,511,1023,8)
2360   fill(0,0,511,29,2)
2370   fill(0,994,511,1023,2)
2380   fill(0,0,23,1023,2)
2390   fill(488,0,511,1023,2)
2400   symbol(168,4,"Christmas Tree",1,1,2,15,0)
2410   box(26,854,137,990,10)
2420   box(140,854,484,990,10)
2430   c_put(30,890,0)
2440   symbol(42,862,"Stock Card",1,1,1,15,0)
2450   symbol(82,918,"のこり",1,1,1,15,0)
2460   symbol(290,862,"捨　札",1,1,1,15,0)
2470   mouse(4):mouse(1)
2480   msarea(0,0,511,511)
2490 endfunc
```

「式」の活用テクニック

tinyCalcの基本技

Izumi Daisuke
泉 大介

一見するとただの小規模な表計算ソフト，行列演算関数がタダ者ではない？ しかしtinyCalcにはまだまだ秘められた機能があります。今回はtinyCalcでプログラム機能を使うための基礎知識について解説します。これによって驚くほど柔軟な処理が可能となることがわかるでしょう。

tinyCalc.xの使い心地はいかがでしょうか。遊んでもらえてますか？ 先月はページの都合もあり，tinyCalcの全機能を紹介することができませんでした。先月号の記事だけではtinyCalcはとおり一辺倒の計算ソフトとなんら変わることがありません。今月はtinyCalcならではのカッ飛んだ使い方を紹介したいと思います。

まずはメニューの使い方を少々

セルの上でマウスの右ボタンをクリックすると，シート最下行にメニューが表示されます。もう一度マウスの右ボタンをクリックするか，ESCキーを押すとメニューはキャンセルされます。メニューの選択は，目的のメニューの上でマウスの左ボタンをクリックするか，メニュー名の最初に書いてある英字をタイプするかで行います。メニュー内ではキーボードによるマウスオペレーションは実行できません（手抜きです）。メニューには以下のものがあります。

●F/ファイル

ファイル入出力を行います。tinyCalcが扱うファイルはCSV形式と呼ばれるデータファイルです。これは，列方向のデータをカンマ(,)で，行方向を改行で区切ったテキストファイルです。tinyCalcでは管理工学研究所が提唱しているK3フォーマットと呼ばれるものを採用しています。これは，数値はベタで，文字列データはダブルクォートでくくるCSV 形式のファイルです。文字列の中にダブルクォートが含まれている場合はダブルクォートを2つ続けることによって表現します。このほか，コメント行なども決められているのですが，tinyCalcではコメント行はサポートしていません。

ファイルメニューには次の5つのサブメニューがあります。

・L/ロード

ファイルを読み込みます。読み込みはA1セルから順に，ファイルにセーブされている項目数，ファイルにセーブされている行数に合わせて行います。

・S/セーブ

ファイルのセーブです。A1～W66のすべてのセルデータを保存します。tinyCalcではデータファイルの拡張子は「.tc」が推奨です。勝手に拡張子を補うことはしませんので，ファイル名は拡張子まで指定してください。

・G/部分ロード

ロード範囲指定付きのファイルロードです。ファイル名選択のあとロードする範囲を尋ねてきますので指定してください。tinyCalcでは表示されている23×30セルまでしか範囲を指定することができません。30行を越える範囲にデータをロードしたい場合は，まずロード範囲の左上のセルのみを指定してください。続いてロード範囲右下のセル指定になります。

「G/部分ロード」はセル幅に対応したデータロードを行います。セル幅を広げたため画面に表示されていないセルにはデータをロードしません。

・P/部分セーブ

セーブ範囲指定付きのファイルセーブです。セーブファイル名設定後，セーブする範囲の指定になります。範囲指定は「G/部分ロード」と同じです。セル幅に対応しています。

・T/テキスト出力

シートに表示されているイメージをそのままファイルにセーブします。tinyCalcは印刷機能を持っていません。これは中途半端な機能をつけてtinyCalcをlargeCalcにしてしまうより，印刷専門のソフトであるワープロを使ったほうがずっと多彩な表現が可能となるからです。このメニューは画面イメージそのままのテキストファイルを出力します。ワープロで読み込んで思う存分装飾を施してください。tinyCalcは，装飾や印字はワープロの仕事と割り切って，計算機能のみに的を絞っています（計算だけをするのにlargeCalcを使う気も起きませんしね）。

●C/複写

指定したセルのコピーを行います。最初にコピー元のセルにアンダーラインをつけておいてから，このメニューを実行します。この機能はセル幅に対応していません。画面から隠されているセルに入っているデータもコピーします。セル幅対応でコピーしたい場合は，「F/ファイル」メニューの「G/部分セーブ」「P/部分ロード」を利用してください。

コピー元とコピー先の指定には，次のバリエーションがあります。

・コピー元がセル，コピー先が範囲

コピー元のセル内容を，コピー先の範囲に複写します。すなわち，コピー先の範囲は，コピー元のセルデータで埋め尽くされることになります。

・コピー元が範囲，コピー先がセル

範囲のコピーを行います。コピー先として指定したセルを左上とする該当範囲に，コピー元の範囲のデータを複写します。コピー元と同じ範囲をマウスで指定するのは煩わしいものです。tinyCalcでは左上のセルだけを指定します。

●E/消去

セルのデータを消去します。個々のセルに入っているデータは，データ入力時にDELキーでデータを削除すれば消すことができますが，複数のセルデータを一気に消したい場合にはこのメニューを利用します。

「E/消去」には，次のサブメニューがあります。

・A/全消去

A1～W66のすべてのセルのデータを消去します。

・E/消去

このメニューを指定すると，消去範囲を尋ねてきます。マウスで消去範囲を指定してください。

●R/更新

tinyCalcでは，マウスの左ボタンクリックでセルにアンダーラインをつけた場合を

除いて，セルの再計算を行いません。したがって，「@sum(A1,D1)」と入力してあるセルの値は，A1～D1セルのデータを変更しても書き換えられないことになります。

このメニューは，セルの自動更新関係の設定を行うメニューです。たとえば上の式を入力したセルを自動更新セルに指定すると，データが入力されるたびに範囲の合計を求め直します。

「R/更新」を選択すると4つのサブメニューが表示されますが，最初の「S/設定」と，残りの「P/全停止」「F/標準」「A/自動」は少し機能が異なっています。後者3つは自動更新モードを設定するためのメニューで，現在選択されているモードは反転表示されています。

・F/標準

最初はこのモードになっています。これはアンダーラインが引かれたセルのデータを更新するモードです。アンダーラインのついたセルのデータを更新したあと，自動更新セルとして設定されているセルのデータを更新します。

・P/全停止

これは，まったく式の計算を行わないモードです。セルに式を入力しても，その結果は計算されません。

・A/自動

これはすべてのセルを自動的に更新するモードです。更新は左から右へ，上から下へ順に行われます。セルにデータを入力するとマウスカーソルが消え，全データの更新を行ったあと再びマウスカーソルが表示されます。もっとも安直に利用できますが，すべてのセルを更新するため再びマウスカーソルが現れるまで時間がかかるのが難点です。マウスカーソルが消えているあいだにキーを押すと自動更新は中断され，自動更新全停止モードになります。これは無限に更新し続けられることがないようにという配慮からです。

・S/設定

自動更新するセルの設定や解除を行います。3つのサブメニューがあり，「S/設定」で自動更新するセルの登録を，「C/解除」で自動更新登録したセルの解除を，そして「A/全解除」で登録されたすべてのセルの解除を行います。

●I/入力方法

範囲を指定してデータ入力を行う際に，縦方向優先でデータを入力するか，横方向優先でデータを入力するかを設定します。初期状態では横方向優先入力になっています。

●Q/終了

tinyCalcを終了します。

＊　＊　＊

いかがでしょうか。自動更新セルを使うと，データを変更して結果を見たい場合に重宝します。まァ一種のシミュレーションに使える，といったところですか。

tinyCalcの扱う式

先月の演算子の説明に，比較とか論理積，論理和などが含まれているのを見て，「あやしい」と思われた方がいらっしゃるかと思います。そうです。tinyCalcはセルの中にプログラムを書き込めるようになっているのです（へへへ，ただの計算用紙ではないぞ）。ここでは，tinyCalcの扱う式について説明します。

プログラムが書き込めるのに「式の説明」とはこれいかに，と思われるかもしれません。tinyCalcはプログラムも「式」として表記します。式であるからには当然値を持つわけで，この値がセルに表示されることになります。

●単式

単式は，数値やセル指定，関数呼び出しを演算子でつないだ一般にいう式です。また，単式の特例として，演算子を含まない項だけのものも単式とします。先月号の説明で使用していたのはすべて単式です。単式の値は，その式を実行した答えです。

●連式

プログラムの最も簡単な形です。単式を，‘；’で区切って複数並べたものをtinyCalcでは連式といいます。たとえば，

A1＝3； A1＊B1

という連式は，A1セルに3を代入したあと，A1×B1を計算します。連式の値は，最

デバッグ情報

締め切りの押し迫る3月中旬，出荷直前になって発見されたバグを修正したところ，かえってエンバグしてしまいました。動作上なんの問題もありませんが，もしよろしければ修正してtinyCalc.xをお楽しみいただければと思います。

エンバグしてしまったのは，ソースファイルのcalc.cです。78行と79行が入れ替わってしまっています。

```
mouse( 0 );
setmspat( );
```

と行を入れ替えてコンパイルし直してください。コンパイルはXCのバージョン2か，GCC＋XCバージョン2のライブラリという構成で行います。バージョン1のライブラリではリンクできません。

「コンパイラもってないんだよ～」という方，「XCのバージョン2は買ってないや」という方のために，修正用のBASICプログラムを用意しました。リストを入力して実行してください。なお，30行は皆さんがtinyCalc.xを格納しているディレクトリにあわせて書き直してください。入力時の注意点は，60～90行のデータ部分とfseek関数の引数です。ここを間違えると解凍からやり直す羽目に陥ります。ご注意ください。

この修正は，マウスでシートをスクロールさせるためのものです。先月はROLL UP，ROLL DOWNキーによるスクロール方法だけを説明しましたが，tinyCalcはマウスでもスクロールさせることができます。シート左の行番号が表示されているところでマウスの左ボタンを押すと，マウスカーソルが上下矢印に変わります。そのまま上にドラッグするとシートも上に，下にドラッグするとシートも下にスクロールします。このとき矢印は，上下どちらかの矢印だけになります。数行分上下にドラッグしただけだとスクロールはゆっくりですが，さらに上下方向にドラッグすると高速スクロール（といってもそれほど速くはないが）するようになります。スクロールをやめるには，ボタンを離すのがてっとり早くていいでしょう。

さらに，X-BASICによる簡単な修正では直せないバグが発見されました。文法チェックを行っている最中にセルの相対指定が有効かどうかをチェックしてしまっているのです。このため，A1セルに

```
a1=1; [a1-1,0]=0
```

という式を入力すると，エラーになってしまいます（初期状態ではA1セルに0がセットされているため，A1セルから[-1,0]セルを操作しようとしたと勘違いしてしまう）。

これをチェックしないようにするには，calculate.cの447行から始まるrelativeAdrs関数内の453行を，

```
if (!PARSE && (*x<1 || 23<*x))
```

に，そして464行を，

```
if (!PARSE && (*y<1 || 66<*y))
```

に変更して再コンパイルしてください。

これからもバグが見つかるたびに修正をなんらかの形で掲載していきます。編集部気付で報告していただけると幸いです。

リスト patch.bas

```
 10 /* tinyCalcのバッチ当てプログラム
 20 /* tinyCalc.xをフルパスで↓に書き込む
 30 str  calc$ = "B:¥calc¥tinyCalc.x"
 40 int  fp
 50 char code( 16 ) = {
 60      &H42, &HA7, &H47, &HF9,
 70      &H0,  &H0,  &HBD, &H36,
 80      &H4E, &H93, &H4E, &HB9,
 90      &H0,  &H0,  &H9B, &HBE
100 }
110 char patch( 6 )
120 /*
130 fp = fopen( calc$, "rw" )
140 fseek( fp, &H192, 0 )
150 fwrite( code, 16, fp )
160 fseek( fp, &HE24C, 0 )
170 fread( patch, 6, fp )
180 patch( 1 ) = patch( 1 ) + 2
190 patch( 5 ) = patch( 5 ) - 2
200 fseek( fp, &HE24C, 0 )
210 fwrite( patch, 6, fp )
220 fclose( fp )
230 end
```

後に計算した式の値です。上の例では，A1×B1が値となり，この連式が書き込まれているセルに格納されます。「A1=3」の値である3はどこにも残りません。

注意してほしいのは連式を単式の中で使うことはできないという点です。連式の最後の式がその値となるからといって，

3*(A1=3； A1*B1)

という表記はできません。

図1

図2

●制御式

一般のプログラミング言語でいう制御構造はtinyCalcでは式の実行を制御するための式，「制御式」として実現されています。制御式も式である以上値を持ちます。通常は最後に実行した式の値が，その制御式の値となります。以下にtinyCalcが用意している制御式について説明しましょう。なお，以後，特に断らない限り，単式，連式，制御式をまとめて「式」と表記します。連式同様，制御式も単式の中で使うことはできませんが，連式の中では使うことができます。

●#if

文法：#if 条件
　　式
　〔#elseif 条件
　　式〕
　〔#else
　　式〕
　#endif

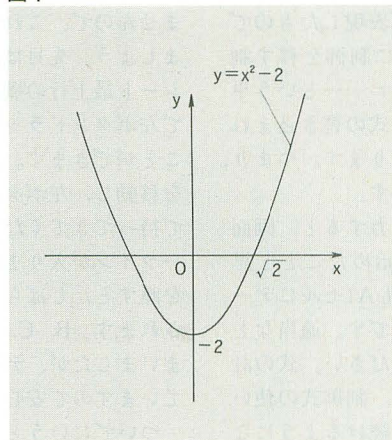
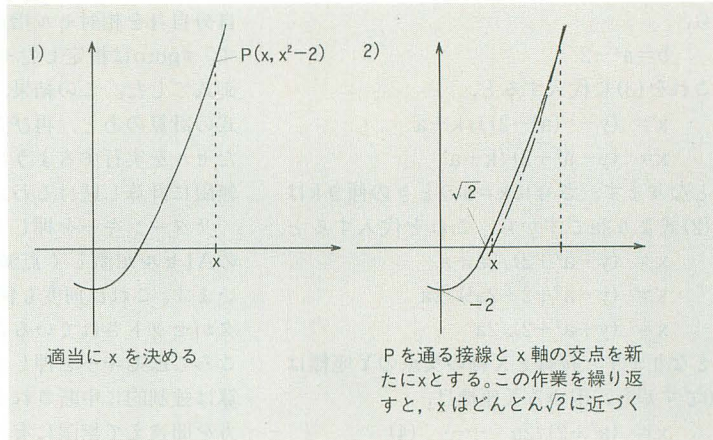
#ifは条件に応じて実行する式を選択します。文法が把握しやすいようにここでは改行を入れて表記しましたが，実際に使うときには改行せず，一気に書き込んでください。〔 〕内は省略可能です。

条件は，単式を論理演算子でつないだ，いわゆる論理式に限られます。ただし，条件にカッコをつけ「(条件)」とすることで，事実上どんな単式も条件となりえます。これは以下の制御式でも同様です。

#if a1=0 b1=1 #endif

はエラーですが，

#if(a1=0) b1=1 #endif

はOKです。この例では，A1セルに入れた値0を#ifの条件として使っています。条件は0が偽，0以外が真ですから，上の例では条件不成立となります。

#if制御式の値は実行した式の値です。条件が成立せず，#elseの指定もない場合には#ifは値なしという値を返します。その結果，#ifが書き込まれているセルの値は変更されません。

●#while

文法：#while 条件
　　式
　#endwhile

条件が成立している間ループを続けます。値は式の値です。

●#repeat

文法：#repeat
　　式
　#until 条件

式を実行し，条件をチェックします。もし条件が偽ならループします。「条件が成立するまで」と考えてください。式の値が#repeat制御式の値となります。

●#goto

文法：#goto セル指定

指定されたセルに制御を移します。この結果，#goto以降に書かれた式は実行されません。指定されたセルを実行した値が#gotoの値となります。

●#gosub

文法：#gosub セル指定

指定されたセルを実行後，その値を持って制御が帰ってきます。冒頭で述べたとおり，制御式を単式の中で使うことはできませんので，値が返されることに大して意味はありません。式を実行する際に，あらかじめアップデートしておきたいセルがある場合に有効です。

●@fnc

文法：@fnc(セル指定)

@fncはその名の示すとおり関数です。特殊な使い方となるため，先月の関数一覧にはあえて収録しませんでした。@fncは指定されたセルを実行し，その値を持って帰ってくる関数です。#gosubは制御式ですので単式の中で使うことができませんが，@fncは関数ですので単式の中で使うことが可能です。これを利用した例は来月解説することにします。

実践編

では簡単なサンプルを示しながら，これら制御式の実際の使い方を見ていくことにしましょう。例題としてニュートン法による平方根の求め方をとりあげます。その前に，まずはニュートン法について説明しておきましょう。

●ニュートン法

図1をご覧ください。これは，

$y=x^2-2$ …… (1)

のグラフです。このグラフがX軸と交わる場所のX座標が2の平方根$\sqrt{2}$になります。問題は，どうやってグラフとX軸の交点を求めるかです。ニュートン法では，図2のような方法を使います。まず，図2-1のように適当にxを決め，グラフに接線を引きます。この接線とX軸の交点を新しいxとし，再びグラフの接線を引きます（図2-2）。以下この作業を繰り返していけば，xはどんどん$\sqrt{2}$に近づいていくことになり，最終的には$x=\sqrt{2}$になるというわけです。

ニュートン法を使うにはまず，点(x,x^2-2)における接線の傾きを知る必要があります。接線の傾きを求めるには微分すればいいだけですから，xにおける接線の傾きkは，

$k=2x$ …… (2)

です。また，点(a,b)を通る傾きkの直線の方程式は，

$y-b=k(x-a)$

となります。これをxについて解くと，

$x-a=(y-b)/k$

$x=(y-b)/k+a$ …… (3)

となります。点(a,b)は(1)式上の点ですから，

$b=a^2-2$

これを(3)に代入すると，

$x=(y-(a^2-2))/k+a$

$x=(y-a^2+2)/k+a$

となります。さらにx=aのときの傾きkは(2)式より2aですから，これを代入すると，

$x=(y-a^2+2)/2a+a$

$x=(y-a^2+2+2a^2)/2a$

$x=(y+a^2+2)/2a$

となります。接線とX軸の交点のY座標は0ですから，求めるX座標は，

$x=(a^2+2)/2a \quad \cdots\cdots \quad (4)$

です。こうして求めたxを新たにaとして(4)式に入れて計算を繰り返せば，xは限りなく$\sqrt{2}$に近づいていくことになります。

●tinyCalcで$\sqrt{2}$を計算してみる

では実際にやってみることにしましょう。ここではA1セルに答えを求めることにします。まず最初に，A1セルに適当な初期値を入れておきます。ここでは2を入れておきます。

B1セルには，

```
a1=(a1*a1+2)/(2*a1)
```

という式をセットします。(4)式では適当に決めたxをaとして記述してありますが，ここでは初期値も，(4)式の計算結果もA1セルに入れることにします。

B1セルに式を入力してリターンキーを押すと，A1セルには新しい答えが表示されました。A1セルでマウスの左ボタンをクリックしてアンダーラインをつけ，シート下に表示された数を確認してみてください。1.5となっています。これは(4)式を一度だけ実行したときの値です。B1セルにマウスカーソルを移し，クリックしてアンダーラインをつけてみてください。A1セルの値が変わりました。再びA1セルをクリックして数値を確認してみましょう。今度は1.4166666666667と表示されたはずです。先ほどの1.5をaとし，再び(4)式を実行した結果です。このようにA1，B1と交互にクリックしていくことで，数値がどんどん$\sqrt{2}$に近づいていくのが確認できます。

tinyCalcではこのように，アンダーラインがつけられるときに再計算が行われます。上の例のように$\sqrt{2}$に近づいていく様子を確認できるのも面白いのですが，答えだけがほしいときにいちいちクリックするのは面倒です。こんなときに制御式が役に立ちます。B1セルの式を，

```
a1=(a1* a1+2)/(2* a1) ;#goto
[0,0]
```

に変更してみましょう。[0,0] というのは自分自身を相対セル指定で表現したものです。#gotoは指定したセルに制御を移す制御式でした。この結果，a1=……という単式の計算のあと，再びこの式の書き込まれたセルを実行するようになります。つまり，無限に計算し続けるわけです。

リターンキーを押して入力すると，画面のA1セルが激しく点滅し始めたことと思います。これは何度も何度もA1セルにデータがセットされているためです。適当なところでESCキーを押してください。式の計算は強制的に中断されます。制御式の使い方を間違えて無限に実行し続けるようになってしまった場合にも，ESCキーは有効です。ESCでなくともいいのですが，ESCキーなら中断後メニュー表示になりますので便利です。というのは，キーボードで計算を強制終了した場合，「再計算」メニューの再計算モードが「全停止」になっているからです。

●収束したら中断する

答えが無理数の場合，(4)式を無限に繰り返さなければ正しい答えは得られませんが，計算機は無限桁扱えるわけではありませんので適当なところで中断するのが賢明です。中断の条件は，(4)式でx=aとなったときというのがいいでしょう。

等しくなったかどうかを判定するには，#if制御式を使います。B1セルは次のようになります。

```
a2=a1;
a1=(a1*a1+3)/(2*a1);
#if(a1!=a2)
  #goto [0,0]
#endif
```

読みやすいように行を変えて書いていますが，入力するときは続けて書いてください。ここでは，$\sqrt{3}$の値を求めてみました。A2セルにA1セルの値を書き込んでおき，A1セルに新しい答えが求まった時点でA2セルに保存しておいた値と比較。同じでなければ#gotoで再計算するようにしています。#gotoの後ろに‘；’がない点に注意してください。これは条件成立時の処理が単式だからです。‘；’は複式で単式や制御式を区切る場合にだけ使用します。

●収束過程を見る

上の例をさらに発展させて，収束過程を目で見られるようにしてみましょう。また，$\sqrt{2}$の計算を$\sqrt{3}$の計算に変えるのに式を修正しなければならないというのもしゃくなので，平方根を求める数をセルで指定するようにもしてみます。

いまのままでは4桁しか見ることができませんので，これを拡大することから始めましょう。先月は説明しませんでしたが，シート最上行の欄名が表示されている場所で左ボタンドラッグすると，欄幅を広げることができます。Aの上にマウスカーソルを移動し，左ボタンを押したままDの上まで持ってきてください。欄名の下にアンダーラインが入りましたね。ここで左ボタンを離すと，しばらくの沈黙のあと欄が拡大されます。B，C，D欄が画面から消えてしまいましたが，データはちゃんと保存されていますので安心してください。

ついでにいうと，欄名の場所でマウスの右ボタンをクリックすると，表示する数値のフォーマットを設定できます。フォーマットの指定は@prnt関数の第2引数と同じ要領です。

フォーマットを設定すると，同じフォーマットを設定する範囲を尋ねてきます。ここではフォーマットを設定する範囲の左端の欄名と，右端の欄名を「AW」のように続けて設定します。設定の解除は解除したい欄の上で右ボタンクリックし，表示形式，設定範囲のいずれにもリターンキーのみを押せばOKです。範囲を指定して設定を解除することはできません。

A欄をD欄の場所まで広げたら，まずG1セルに平方根を求めたい数を書き入れます。そしてF1セルに次の式を書き込んでください。

```
f1=0;
a1=g1;    #repeat
    [-5,f1+1]=([-5,f1] * [-5,
f1]+g1)/(2* [-5,f1]);
   f1=f1+1    # until [-5,f1-
1]== [-5,f1]
```

相対セル指定になっているのでわかりづらいかもしれませんが，[-5,?]というのはA?セルのことです。一度実行してみていただいたほうがわかりやすいかもしれませんね。やっていることは，F1セルに相対行数を入れ，F1を増やしながら先ほどの計算を繰り返しているだけです。G1セルにほかの数をセットして計算させる場合は，いったんA欄を「消去」メニューの「消去」でクリアしたほうが，収束点がわかっていいと思います。

*　　*　　*

今回は関数を組み合わせたプログラムの作り方を解説しました。来月はさらに応用例と自分自身でtinyCalcの関数を作成していく際に必要となる情報を公開していく予定です。お楽しみに。

MAGICの拡張

先月は前半でMAGICの使い方を簡単に説明し，後半はX68000版になって拡張された部分の話をしました。今月は先月紹介できなかったMAGICに従来からあるコマンド群の説明，および，プログラムを改良する場合にはどうすればいいかについて解説していきます。

Kageyama Hiroaki
影山 裕昭

MAGICのコマンド

先月はコマンドを表にまとめて掲載しました。あの表さえあればひととおりMAGICのコマンドが使えるはずですが，先月のおさらいも含めてもう一度説明しておきましょう。もともとのMAGICは$00〜$0Fの16種類のコマンドで構成されていました。これらのコマンドをざっと紹介します。

●$0000　LINE

グラフィック画面に直線を引くコマンドですが，正確にはコネクトラインとでも呼べるもので，与えられた座標を順番に直線で結ぶコマンドです。つまり，座標1〜座標4を与えると，座標1と座標2を結ぶ直線を引き，次いで座標2と座標3を結ぶ直線を引き，最後に座標3と座標4を結ぶ直線を引きます。つまり一筆書きですね。IOCSコールLINEを使って描画してます。

●$0001　SPLINE

3点を通るスプライン曲線を描きます。掲載したプログラムはちょっと見てもわかるほど，いろいろと無駄の多いものになっています。ライン描画はMAGICのコマンドLINEを使っています。

●$0002　BOX

2点を対角線とする長方形を描画します。IOCSコールBOXを使っています。

●$0003　TRIANGLE

3点を頂点とする三角形を塗り潰して描画します。SPLINEと同様に無駄の多いプログラム。さらに悪いことに，バグが見つかってしまいました。サブルーチンraster(IOCSコールLINEを使って水平線を引く）を使って横ラインを引きます。

●$0004　BOX FULL

2点を対角線とする長方形を塗り潰して描画します。IOCSコールFILLを使っています。

●$0005　CIRCLE FULL

内部を塗り潰した円を描画します。このプログラムも最適化がまったくされていません。TRIANGLEと同じくrasterを使っています。

●$0006　SET WINDOW

2点を対角線とする長方形を表示範囲とします。IOCSコールWINDOWを使ってIOCSのワークエリアにウィンドウ座標を設定するとともに，MAGIC内部のワークエリアにも座標値を格納します。

minx:　左上x座標
miny:　左上y座標
maxx:　右下x座標
maxy:　右下y座標

●$0007　SET MODE

ラインの描画モードを設定します。もともとはXOR，表示しない，というモードもあったのですが，現在サポートしているのはPSETとPRESETの2つの描画モードです。XORがないのはMAGICが利用しているIOCSコールのラインルーチンにXORのモードがないからです。また，表示しないモードはあまり必要性がないと考え，現在サポートしていません。

●$0008　POINT

指定座標のカラーコードを調べます。IOCSコールPOINTを使っていますが，このコマンドを使うくらいなら，直接IOCSコールPOINTを呼び出したほうがいいと思います（無責任なコマンドだ)。

●$0009　CLS

ウィンドウ内をクリアします。先にCRTで画面モードを設定をしておけば，このコマンドを実行しなくてもグラフィック画面はクリアされています。IOCSコールWIPEを使っています。

●$000A　RESERVED

もともとはパレットを変更するものだったのですが，X68000版MAGICではこのコマンドを指定しても無視されます。

●$000B　SET 3D PARAMETER

3D-2D変換に必要なパラメータを設定します。

●$000C　SET 3D DATA

頂点，線分のつながりで3D物体の形状を設定します。

●$000D　TRANSLATE 3D→2D

3D物体を3Dパラメータに従って2Dに変換します。サブルーチンsincosを使ってsin値，cos値を求めています。

●$000E　DISPLAY 2D

3D-2D変換した物体を表示します。ここでもライン描画はIOCSコールLINEを使っています。

●$000F　DONE

MAGICを呼び出したシステムに制御を戻します。

コマンドを直接呼び出す

MAGICのコマンド列を呼び出すには，a0にコマンド列のアドレスをセットしてAUTOコマンドを実行すればいいことを先月説明しました。画面をクリアするなら，

```
clear:
        dc.w    9
        dc.w    $0F
```

というコマンド列を用意して，

```
__AUTO: equ  $FD13
        lea.l  clear,a0
        dc.w   __AUTO
```

とすればいいのですね。しかし，ひとつのコマンドを実行するのにAUTOコマンドを使うのもなんなので，

```
__CLS: equ  $FD09
```

とラベル定義しておき，

```
     dc.w   __CLS
```

とすることもできます。この呼び出し方はコマンド番号に$FD00を足した値で各コマンドを個別に呼び出す例ですが，普段はなるべくAUTOコマンドを使うようにしてください。参考までにラインをこの方法で呼び出すと次のようになります。

```
__LINE:  equ   $FD00
        lea.l  sample,a0
        dc.w   __LINE
        dc.w   $FF00      *DOSに戻る
```

```
sample:
        dc.w  2          ＊座標総数
        dc.w  0,0        ＊座標1
        dc.w  767,511    ＊座標2
```

コマンド列の先頭にコマンド番号を記述しないことと，最後に$000F (DONE)をつけなくていいことに注意してください。また，プログラムは高速化のためにエラーチェックをまったく行っていないので，$0～$13($FD00～$FD13)以外をコマンド番号を設定すると確実に暴走します。

また，

```
    dc.w  $FD0F
```

は自殺行為です。ソース(magic.s)を見てもらえば理由はわかると思います。

コマンドの改良について

X68000版MAGICは1986年9月号に掲載された8ビット版MAGICを参考にして作られています。移植作業はZ80のコードを68000のコードに置き換えることで行われました。グラフィック描画をIOCSコールに依存すれば，すべてのコマンドが単純作業で動作することが予想されたからです。結果は思惑どおりで，編集室に依頼されてから3D物体が動くようになるまでの開発期間は自分でも驚くほど短いものでした。

その後，3D表示関係のコマンドについて，機能拡張および1クロックでも速くなるようにコードの見直しに取りかかりました。機能拡張部分は先月説明したバッファを2つ持たせた点や，画面切り替えを行うようにした点で，マスターアップ前の約1カ月間はこのために費やされました。そのため，3D関係については，ある程度68000らしいプログラムになったのですが，その他のコマンドがZ80のバカ移植のままでマスターアップとなってしまいました。

コマンド説明にもあるように，SPLINE，TRIANGLE,CIRCLE FULLのプログラムは改良の余地がかなり残されています。おまけにTRIANGLEはバグまで出てしまいましたので，この場でTRIANGLEを改良したものを掲載し，コマンドを改良する際の参考にしてもらいたいと思います。

図1はディスクに収録されたプログラムが，どのコマンドに対応しているか表したものです。今回はTRIANGLEを改良するので，TRIANGLE.Sを変更することになります。まずはTRIANGLE.S改良版をリスト1に紹介しておきますので，説明とあわせてご覧ください。

まず，コマンドを作るうえで必要となるパラメータの取り込み方法を説明します。プログラムの入り口でa0の値が意味を持ちます。

a0.l　パラメータの置かれたアドレス

TRIANGLEの場合，サイズがワードのパラメータが6つありますから，プログラムの先頭(23行)で，

```
    movem.w  (a0)+,d1-d6
```

として，x1,y1,x2,y2,x3,y3の値をd1～d6に取り込みます。必要ならばワークにも値を格納しておきましょう。

さて，入り口の話をしたついでに出口の話もしておきましょう。MAGICで意味を持つ戻り値は，こちらもa0のみで，

a0.l　次のコマンド

となっています。このため，プログラムでは必ずa0の値をパラメータのバイト数だけ増やして終了するようにします。忘れるとAUTOコマンドが正常に動かなくなりますから，細心の注意をはらってください。普通は23行のようにパラメータをまとめてポストインクリメントアドレスレジスタ間接形式を使って取り込み，あとはa0を使わないようなプログラムを組めば問題ありません。どうしてもa0を使いたい場合はスタックなどに値を退避し，リターン時に復帰させるなどの対処策をとってください。

次に，コマンドのエントリラベルを定義します。これは表2のようになっています。コマンド番号3のTRIANGLEは図2からTRIANGLEがエントリラベルなので，忘れずに外部定義しておきます(9，22行)。コマンドを改良する際は以上の点にさえ注意すれば大丈夫なはずです。

新たにコマンドを拡張する場合はMAGIC.Sに表2のような部分がありますから，

```
    dc.l  auto
```

の下に，

```
    dc.l  new_command  ＊ 14
```

を追加し，さらにプログラムの先頭で，

```
    .xref new_command
```

として外部参照プログラムとします。新たに作成するプログラムにも，

```
    .xdef new_command
```

と外部定義しておき，リンク時に既存のファイルに加えます。

コマンド募集

コマンドの改良・拡張は，制作者である私はともかく，第三者の読者の皆さんにとっては，MAGICへの組み込み方など，わかりづらい面が多いと思います。それから，説明したあとでこういうのも変ですが，コマンドの改良・拡張については極力私が行うようにします。どうしても改良してみたいんだけど，よくわからない部分があったら，私がOh!X質問箱を担当していますので，そちらのほうに質問を送ってください。また，ほしい機能などありましたら編集室影山宛てにお願いします。アンケートはがきに書いてもらってもいいです。

私から読者の皆さんに制作をお願いしたいのはグラフィックルーチンです。たとえば，高速BOX，高速FILL，高速WIPE，特に高速LINEを募集します。仕様としては，ラインスタイルは無視して結構ですが，クリッピングはやってください。その他，引数の受け渡し方法は，とりあえずIOCSコー

図1

```
¥magic               * コマンド名
|-BOX.S              * BOX
|-BOXFULL.S          * FILL
|-CIRCLE.S           * CIRCLE
|-DATA.S             * 3D_DATA
|-DISP_FLAME.S       * DISPLAY 2D
|-INIT.S             * INIT
|-LINE.S             * LINE
|-MAGIC.S            * MAGICの常駐処理、AUTO
|-MODE.S             * MODE
|-PARA.S             * 3D_PARA
|-PERSPECTIVE.S      * 3D_TRANS
|-POINT.S            * POINT
|-RASTER.S           * 水平線を引く
|-SCRMOD.S           * CRT
|-SET_COLOR.S        * COLOR
|-SINCOS.S           * SIN,COSの値をテーブルから求める
|-SPLINE.S           * SPLINE
|-TRIANGLE.S         * TRIANGLE
|-WINDOW.S           * WINDOW
|-WIPE.S             * CLS
|-MAGIC.MAC          *
|-MAGIC.H            *
|-WORK.H             *
|-MAGIC.X            * MAGIC本体
|-SAMPLE.S           * 四角枠を動かす
|-TYRREL.S           * ???を動かす
|-MAKE.BAT           * MAGIC.X作成バッチファイル
```

図2

```
jmptbl:                              * コマンド番号
        dc.l    line                 *  0
        dc.l    spline               *  1
        dc.l    box                  *  2
        dc.l    triangle             *  3
        dc.l    boxfull              *  4
        dc.l    circle               *  5
        dc.l    window               *  6
        dc.l    mode                 *  7
        dc.l    point                *  8
        dc.l    cls                  *  9
        dc.l    no                   *  a
        dc.l    para                 *  b
        dc.l    data                 *  c
        dc.l    perspective          *  d
        dc.l    disp_flame           *  e
        dc.l    done                 *  f
        dc.l    set_color            * 10
        dc.l    scrmod               * 11
        dc.l    init                 * 12
        dc.l    auto                 * 13
```

図3

```
raster.s
機能：水平線を引きます
入力：d1.w  始点x座標
      d2.w  終点x座標
      d3.w  y座標
出力：d0.l  0 正常終了
            －1 グラフィック使用不可
  (ただし，MAGICではd0を参照しません)
レジスタ：a1破壊
```

ルコンパチにお願いします。5月号で村田敏幸氏が作成したラインルーチンはIOCS.Xの約2倍だそうですので，これよりも高速なものを期待しています（もちろん，今月号のよりも速いものを）。

もうひとつ，改良しやすいものとしてraster.sがあります。このサブルーチンの仕様を図3に示します。興味のある方は改良に取り掛かってみてください。

MAGICへの組み込み方がわからなければ，私のほうでインストールします。できるだけ多くの皆さんからの投稿を待っています。

おっと，MAGICに新しいTRIANGLEを組み込む方法を説明していませんでした。リスト1を入力したら，付録ディスクのTRIANGLE.Sにコピーして，アセンブル，リンクします。MAGIC.Xの作成は付属のMAKE.BATを使うと便利です。

今後の予定

現在MAGICをX-BASICの外部関数として使えるようにMAGIC.FNC(仮称)を制作中です。

実は今月発表したTRIANGLE.SはMAGIC.FNC用に制作されたプログラムなのです。ソースにもあるように，1本のプログラムでMAGIC.X，MAGIC.FNC，MAGIC LIBに対応させる予定です。7月号発表をめざして頑張っていますので，ご期待ください。

MAGICプログラム大募集

もちろん募集しているのはMAGICの内部ルーチンだけではありません。MAGICを使ったアプリケーションも募集中です。S-OS上には2D系コマンドを使ったアニメーションツールなんてのもありましたね。いまはまだ「高速グラフィックパッケージ」というほど高速ではないのですが，今後はちゃんと速くなることが予想されます。ということで皆さん，MAGICを使ったアプリケーションもよろしく。

リスト1

```
  1: *
  2: * triangle.s    version 1.15
  3: *
  4:
  5:         .include        iocscall.mac
  6:         .include        work.h
  7:
  8:         .xdef           _triangle
  9:         .xdef           triangle
 10:         .xdef           triangle_fnc
 11:         .xref           raster
 12:
 13:         .text
 14:
 15: _triangle:
 16:         movem.l d3-d7/a3-a6,-(sp)          * ┐Cライブラリ用
 17:         movem.l 4*9+4(sp),d1-d7            * |
 18:         move.w  d7,line_data+8             * |
 19:         bsr     triangle_fnc               * |
 20:         movem.l (sp)+,d3-d7/a3-a6          * |
 21:         rts                                * ┘
 22: triangle:
 23:         movem.w (a0)+,d1-d6
 24: ************************************
 25: * 頂点の座標(x1,y1)(x2,y2)(x3,y3)を
 26: *     y1≦y2≦y3
 27: * の関係を満たすようにする
 28: ************************************
 29: triangle_fnc:
 30:         cmp.w   d4,d6
 31:         bcc     triangle2       * y3≧y2
 32:         exg     d3,d5
 33:         exg     d4,d6
 34: triangle2:
 35:         cmp.w   d2,d6
 36:         bcc     triangle3       * y3≧y1
 37:         exg     d2,d6
 38:         exg     d2,d4
 39:         exg     d1,d5
 40:         exg     d1,d3
 41: triangle3:
 42:         cmp.w   d2,d4
 43:         bcc     triangle4       * y2≧y1
 44:         exg     d1,d3
 45:         exg     d2,d4
 46: ************************************
 47: *       三角形を塗り潰す
 48: ************************************
 49: triangle4:
 50:         movem.w d1-d6,x1
 51:         move.w  d1,sx
 52:         move.w  d1,ex
 53:         move.w  #$5240,triangle14         * addq.w #1,d0
 54:         sub.w   d1,d3
 55:         beq     triangle4_1     * x2-x1=0
 56:         bpl     triangle5       * x2-x1>0
 57:         neg.w   d3
 58:         move.w  #$5340,triangle14         * subq.w #1,d0
 59:         bra     triangle5
 60: triangle4_1:
 61:-        move.w  #$4e71,triangle14         * nop
 62: triangle5:
 63:         move.w  d3,triangle12+2
 64:         move.w  #$5240,triangle18         * addq.w #1,d0
 65:         sub.w   d1,d5
 66:         beq     triangle5_1     * x3-x1=0
 67:         bpl     triangle6       * x3-x1>0
 68:         neg.w   d5
 69:         move.w  #$5340,triangle18         * subq.w #1,d0
 70:         bra     triangle6
 71: triangle5_1:
 72:         move.w  #$4e71,triangle18         * move.w #$4e71,d0  $4e71 = nop
 73: triangle6:
 74:         move.w  d5,triangle16+2
 75:         move.w  y1(pc),py
 76:         move.w  y3(pc),d1
 77:         sub.w   y1(pc),d1       * y3-y1
 78:         move.w  d1,triangle17+2
 79:         move.w  y3(pc),d6
 80:         sub.w   y2(pc),d6       * y3-y2の値 HL' D6.w
 81:         move.w  y2(pc),d5
 82:         sub.w   y1(pc),d5       * y2-y1の値 DE' D5.w
 83:         move.w  d5,triangle13+2
 84:         tst.w   d5
 85:         bne     triangle7
 86:         move.w  x2(pc),sx
 87:         tst.w   d6
 88:         bne     triangle8
 89:         move.w  x1(pc),d1
 90:         move.w  x2(pc),d2
 91:         move.w  y1(pc),d3
 92:         jsr     raster(pc)
 93:         move.w  x2(pc),d1
 94:         move.w  x3(pc),d2
 95:         move.w  y1(pc),d3
 96:         bra     raster
 97: triangle7:
 98:         move.w  d5,d1
 99:         asr.w   #1,d1
100:         move.w  d1,r_1              * y2-y1/2
101:         move.w  triangle17+2,d1
102:         asr.w   #1,d1
103:         move.w  d1,r_2              * y3-y1/2
104:         subq.w  #1,d5
105:         bsr     triangle10
106:         tst.w   d6
107:         beq     triangle_end
108: triangle8:
109:         move.w  x2(pc),sx
110:         move.w  d6,triangle13+2
111:         move.w  d6,d5
112:         asr.w   #1,d6
113:         move.w  d6,r_1
114:         move.w  x3(pc),d1
115:         move.w  #$5240,triangle14         * addq.w #1,d0
116:         sub.w   x2(pc),d1
117:         beq     triangle8_1
118:         bpl     triangle9
119:         neg.w   d1
120:         move.w  #$5340,triangle14         * subq.w #1,d0
121:         bra     triangle9
122: triangle8_1:
123:         move.w  #$4e71,triangle14         * nop
124: triangle9:
125:         move.w  d1,triangle12+2
126: triangle10:
127:         move.w  py(pc),d3
128: triangle11:
129:         move.w  sx(pc),d1
130:         move.w  ex(pc),d2
131:         jsr     raster(pc)
132:         move.w  r_1(pc),d1
133: triangle12:
134:         move.w  #0,d2               * x2とx1の差
135:         sub.w   d2,d1
136:         bpl     triangle15
137: triangle13:
138:         move.w  #0,d2               * y2-y1の値
139:         move.w  sx(pc),d0
140: triangle14:
141:         nop                         * x2>x1なら addq.w #1,d0
142:                                     * x2=x1なら nop
143:                                     * x2<x1なら subq.w #1,d0
144:         add.w   d2,d1
145:         bcc     triangle14
146:         move.w  d0,sx
147: triangle15:
148:         move.w  d1,r_1
149:
150:         move.w  r_2(pc),d1
151: triangle16:
152:         move.w  #0,d2               * x3とx1の差
153:         sub.w   d2,d1
154:         bpl     triangle20
155: triangle17:
156:         move.w  #0,d2               * y3-y1の値
157:         move.w  ex(pc),d0
158: triangle18:
159:         nop                         * x2>x1なら addq.w #1,d0
160:                                     * x2=x1なら nop
161:                                     * x2<x1なら subq.w #1,d0
162: triangle19:
163:         add.w   d2,d1
164:         bcc     triangle18
165:         move.w  d0,ex
166: triangle20:
167:         move.w  d1,r_2
168:         addq.w  #1,d3
169:         dbf     d5,triangle11
170:         move.w  d3,py
171: triangle_end:
172:         rts
173:
174:         .end
175:
```

SX-WINDOW

SX-WINDOW用環境改善ツール

Digital Arajin & SXWHERE

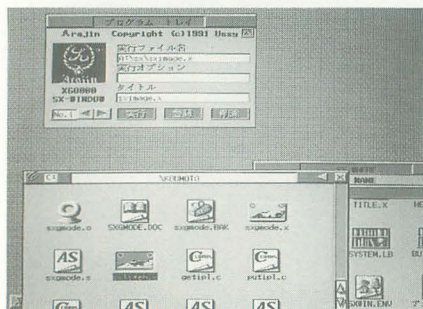
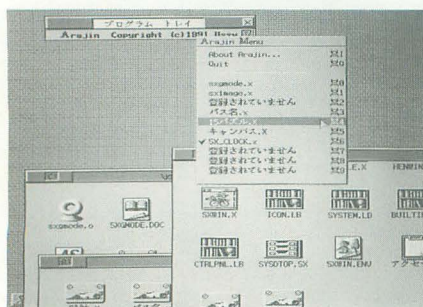
Digital ArajinとSXWHERE，皆さんすでに使っているでしょうか？ 特にハードディスクを使っている場合，これらのツールはSX-WINDOWの操作環境を著しく改善してくれます。今回は先月の使い方では書き切れなかった,さらに細かい使い方について解説します。

Ushijima Takeo
牛島 健雄

SX用実用アプリケーション

Oh!Xの1月号で，SX-WINDOWの開発資料が一般ユーザー向けに発表されてから，はやくも4カ月が経ってしまいました。ネットワークなどでは，そろそろユーザーが開発したSX-WINDOW用アプリケーションも見られるようになってきています。しかし，そこで提供されたアプリケーションの多くは“遊び”を中心としたものです。また，ピンボールや15パズルしか持っていない皆さんもそろそろ遊びに飽きてきた頃ではないでしょうか。ここでは“遊び”を目的としたものではなく，環境を向上させるための2点のアクセサリを紹介します。

プログラムトレイ

プログラムトレイDigital Arajinは，MacintoshのDA（デスクアクセサリ）のように，アプリケーションと登録しておき，好きなときに簡単に呼び出すことができます。

以下に簡単な使い方を示します。

まず，ウィンドウの構成ですが，右上のボタンは，モードの変更に用います。

このモードとは，プログラムトレイが持っている，“実行モード”と“編集モード”という，2つのモードのことです。

実行モードは，登録してあるプログラムを実行するためのモードであり，ウィンドウは小さくなります。起動直後はこの状態です。右上にある“▽”ボタンを左クリックすると，次に説明する編集モードに移行します。

編集モードでは，プログラムの登録・削除/実行ができます(実行モードに戻るときは“△”ボタンを左クリックしてください)。

実際に登録できるプログラムの数は10個までとなっており，それぞれ0～9までのコードが割り当てられています（これをプログラムコードと呼びます)。

ユーザーはひとつのプログラムコードに，ひとつのプログラムを自由に設定することができます。登録したプログラムはプログラムトレイの終了時にファイルとしてセーブされ，次回起動時に自動的に読み込んで設定します。

編集中のプログラムコードは，左下に表示されていて，コントロールで簡単に変えることができます。また，メニューから選ぶ，もしくは“Opt.1＋0～9”を押すことで直接指定することもできます。

これら，プログラムを登録する方法については，後ほど詳しく説明します。

次に，右側の縦に3つ並んだ細長い白い部分ですが，これらは書いてあるとおり“実行ファイル名”，“実行オプション”，“タイトル”を表示する場所です。

“実行ファイル名”

その名のとおり，起動するファイル名を書きます。最大で90文字となっています。

“実行オプション”

プログラムを起動するときに，指定するオプションを125文字以内で指定できます。

“タイトル”

右ボタンを押してメニューを出していただければわかると思いますが，登録されているプログラムは，メニューからも選択することができます。このとき，プログラムの内容が簡単にわかるような簡易説明を最大20文字まで設定できます。登録されていないコードは「登録されていません」というタイトルになっていますので，まぎらわしいタイトルを設定しないようにしましょう。

これらを設定する場合は，それぞれの白い部分を左クリックすれば，カーソルが出ますので，キーボード（または，クリップボード）から入力してください。

残りの右下の3つのボタンは編集のときに使うボタンです。

●プログラムの登録

登録できるプログラムが10個であることは，先ほどの説明で述べたとおりです。さて，これらのプログラムを登録するには，2種類の方法があります。

どちらの方法で登録する場合にも，登録するプログラムコードを先に選んでおいてから行ってください。

1）登録したいプログラムのアイコンをドラッグしてきて，プログラムトレイの上で離すと，実行ファイル名の部分にフルパス名が入力され，タイトルがファイル名になります。

アイコンを放り込んだところ

メニューから……

プログラムを起動

あとは，実行ファイル名からフルパス指定を削ったり，タイトルを変更したりなど，各人の自由に設定してください。

2) 実行ファイル名，実行オプション，タイトルをキーボード（またはクリップボード）から入力する。

上記のどちらかの方法で入力が終わりましたら登録ボタンを押します。

プログラム名が異常な場合にはエラーとなりますが，それ以外では「このまま登録しますか」と聞いてきますので，確認すれば登録します。登録後は，設定内容に間違いがないかどうか確認するために，実行ボタンを押して起動できるか確認することをおすすめします。

異常なプログラム名は，実行形式でないもの（拡張子がX，R，Z，BAT以外のもの）と規定していますので，登録することができません。

●プログラムの削除

登録されているプログラムを，削除するには，削除ボタンを押してください。確認を取ったあとに削除します。削除後のタイトルは「登録されていません」となります。

●プログラムの実行

登録しておいたプログラムを実行するためには，実行モードで選択するか，編集モードで実行ボタンを押してください。

実行モードで起動するためには，メニューから選択するか，ショートカットつまり，Opt.1＋プログラムコードで選択します。

編集モードでは，プログラムコードを選択してから実行ボタンを押してください。

通常は，実行モードで起動するようにします。

●その他の機能

このプログラムトレイでは，ショートカット（Opt.1＋？）を多用していますので，マウスで直接クリックするよりも簡単に操作できる部分もあります。以下にショートカットの機能を示します。

W…モードの変更。押すごとに切り換わります。

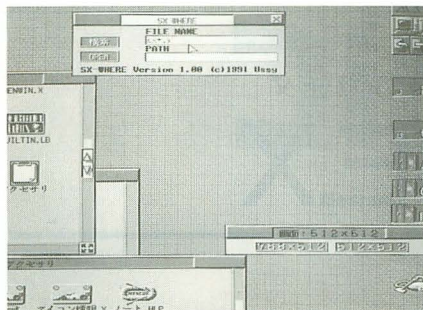
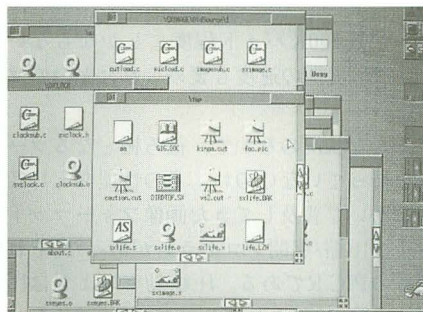
↑…編集モードから実行モードに移行します。

↓…実行モードから編集モードに移行します。

←…プログラムコードを－1します。

→…プログラムコードを＋1します。

以上で，プログラムトレイの主な説明は終わりですが，最後に注意していただきたい点をいくつか述べます。

まず，データファイルは，プログラムの終了時（全クローズ時も含む）にのみ保存されます。そのため，新たに登録したあとに，そのままリセットしてしまうと，保存されませんので，面倒ではありますができるかぎり，一度終了するようにするか，もしくはリセットをせずに，システムの終了を実行してください。

次に，データファイルの場所ですが，これは本体プログラムと同じディレクトリにあることが必要です（セーブ時は必ず同じディレクトリに作成します）のでむやみに移動しないようにしてください。

探したいファイル名をセット

検索中

該当するすべてのディレクトリを開く

パス名でオープン

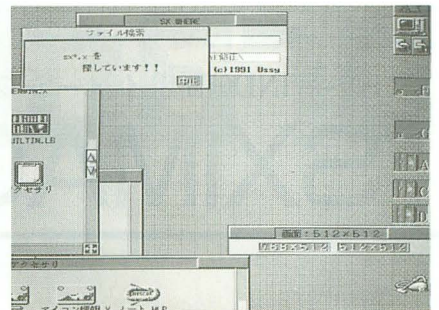
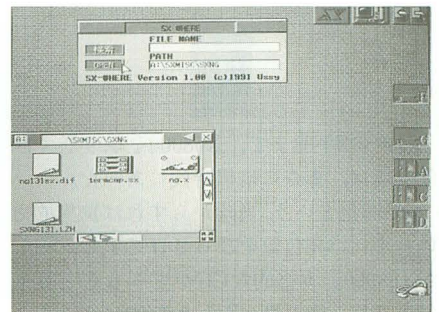
SXWHERE

このプログラムは，Human68kでのWHEREと同じ目的のプログラムなのですが，検索ファイルが存在するディレクトリを，次々に開いていくのが特徴です。

また，特殊な使い方として，簡易ディレクトリオープナーという使い方もあります。

では，使い方を説明します。

検索するファイル名を入力して，検索ボタンを左クリック，もしくは，メニューで選択します。

すると，検索中のダイアログウィンドウが開き，検索を始めます。ファイルが見つかれば，“PATH”という部分に順次表示され，全ドライブを探したところで，見つかったディレクトリをすべて開いて終了します。見つからなかった場合には，その旨のダイアログを開きます。

検索ファイルの入力の方法ですが，“FILE NAME”と書いてある下の白い部分を左クリックするとプロンプトが表示され，入力待ち状態となります。この状態から抜けるにはリターンキーを押すか，ウィンドウのほかの部分を左クリックしてください。

検索ファイルを入力せずに検索を始めると，おかしな動作を行うことがありますので，このような行為はできるかぎり避けてください。

さて，簡易ディレクトリオープナーとしての使い方ですが，検索ファイル名を入力するときと同様に，“PATH”と書いてある下の白い部分を左クリックして，入力待ち状態にしましたら，開きたいディレクトリ名を入力していきます。

入力しましたら，検索ボタンの下のOPENボタンを左クリックするか，メニューから選択すると，そのディレクトリをオープンします。存在しないディレクトリを指定した場合はなにも起こりません。

また，このプログラムでも，Opt.1キーによるショートカットを用いています。検索がOpt.1＋Sキー，オープンがOpt.1＋Oキーとなっています。

＊　＊　＊

以上2つのプログラムを紹介してきましたが，皆さんの操作環境のお役に立てばうれしいですね。

ウィンドウのプログラムは，アイデアさえまとまってしまえば，あとは比較的楽に作れますので，皆さんも頑張ってみてはいかがでしょうか。

SX-WINDOW用イメージファイルローダ

SXIMAGE.X

Tan Akihiko
丹 明彦

SX-WINDOWのデスクトップを彩るグラフィックウィドウ。これまではPIX形式(一部ではMKI, PIC形式も)しか扱えませんでしたが，PICファイルの16色表示とメモリを圧迫しないCUT形式がサポートされました。もちろん邪魔なときはアイコン化も可能です。

状況

ほんの2年前まで，X68000を取り巻く画像環境は，まさに暗黒時代にあった。

65536色という，当時からすれば驚異的といっていい発色数をひっさげてデビューしたX68000。本格的なCGができるぞと誰もが思った。僕も思った。熱狂的に迎えられた65536色。しかし年月を経ずして，その暗い面も明らかになりつつあった。キーワードは「512Kバイト」。とにかく大きいのだ，G-RAMの容量が。そのままセーブすると，画像ファイルは巨大なものになる。比較的大容量のはずの2HDフロッピーディスクをもってしても2枚しか格納できない。

X68000のデビュー後，それぞれのソフトハウスが独自に画像圧縮フォーマットを決めて使いつつあった。これらは画像フォーマットの乱立を招き，若干の混乱を引き起こした。あああMacintoshやAMIGAって偉大だ。メーカー自らが画像フォーマットを初めから提唱してマシンを発表したのだから。

で，それらのフォーマットのなかで現在も生き残っているといえるのは，Z'sSTAFFの標準フォーマットであるZIMファイルくらいのものだろう。これはZ'sSTAFFのツールとしての優秀さと普及率のゆえである。肝心の画像ファイルとしてのパフォーマンスはどうかというと，これはいまひとつ。多少の圧縮はするものの，それでも512Kバイトが数100Kバイト程度にしかならないし，圧縮/展開の速度も褒められたものではない。

PICファイルの表示例

そんな状況のなかで，後ろ楯となるツールを持っていないのに，その性能ゆえに猛烈な勢いで普及してきた画像フォーマットおよび圧縮/展開プログラムがあった。それが噂のPIC.Rである。圧縮効率のよさは圧倒的。512Kバイトがたったの5Kバイトに圧縮されることもある。圧縮/展開の速度もかなりのもの。たいていの絵なら数秒で展開してしまう。実行ファイルの大きさも驚くほど小さい(5Kバイト弱)。「稲妻走る」と呼ばれるアルゴリズムは簡単で(しかし思いつくのは難しい。発想がいいのである)，シンプルイズベストを地でいっている。

そんなこんなでPICがX68000の標準画像ファイルの地位を固めたころ，X68000は新しい環境の登場を見る。SX-WINDOWである。SX-WINDOWはX68000のウィンドウシステムに恥じない仕様として，グラフィックを表示するウィンドウを開くことができるのであった。ディレクトリのウィンドウに混じってグラフィックウィンドウが開いている姿はなかなか新鮮であった。

ところが困った問題が持ち上がった。見え見えの展開ながら，画像ファイル問題である。SX-WINDOWの画像表示プログラムはキャンバス.Xである。このキャンバス.Xは，PIXファイルと呼ばれる独自フォーマットの画像ファイルを読み込んでウィンドウに画像を表示する。SX-WINDOWは(標準では)768×512ドット，16色モードで動作するので，PIXファイルもそれに合わせた仕様になっている。

このPIXファイルの問題は，ほかの画像ファイルとの互換性がまったく取れていないというところにある。ファイルを解析したら，水平型で格納してあることがわかった。X68000のG-RAMは垂直型の構成をとっているのだから，PIXファイルの格納形式は謎といわねばならない(ちなみにキャンバス.Xを覗くと，内部ではしっかり垂直型に変換して格納/表示していることがわかった。これはもう「歴史的理由」以外の何物でもないのだろう)。

おまけに圧縮をまったくしていないので，フルサイズの絵だと実に150Kバイトにも達する。対応するグラフィックツールも現時点ではない。結果として，SX-WINDOWは画像が表示できるけれども，肝心の画像ファイルについてはまったく蓄積がないという状況を招いた。

*

そこで今回発表するSXIMAGE.X。これはSX-WINDOW上で動作する画像表示アプリケーションである。SXIMAGE.Xは，

・CUTファイル
・PICファイル

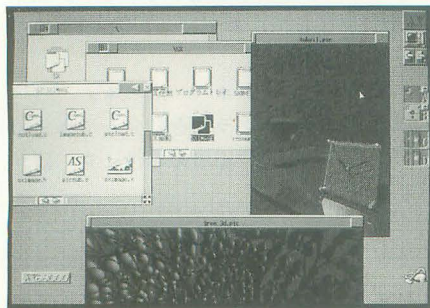
の2種類の画像ファイルを表示する。CUTファイルは白黒2色の画像で，主にドキュメントに埋め込む挿絵として使われる。CUT，PICともに電脳倶楽部や本誌の付録ディスクでお馴染みのはずだ。

本誌付録ディスクの標準シェルとなったVS2.Xにはこの両方を表示する機能がある(VSCUT.X およびVSPIC.X)が，SX-WINDOWでもこれらの画像を見ることができるようになったわけだ。いずれはなんでも表示できるようにとSXIMAGEと名づけた。

SXアプリケーションは簡単でなくちゃ

使い方はいたって簡単。キャンバス.Xに準ずる。

・SX-WINDOW上でSXIMAGE.Xのアイコンをダブルクリックする。
・"SXIMAGE"というタイトルのウィンドウが開くので，表示させたいCUTファイルまたはPICファイルのアイコンをドラッグして，そのウィンドウに放り込む。するとその画像が表示される。

PICファイルの処理には多少時間がかかる(数秒～数10秒)。これは不可抗力の面10%，僕の手抜き90%である。

・画像が表示されているSXIMAGE.Xのウィンドウに別の画像ファイルを放り込むと，新しい画像が表示される。PICのあとにCUTを放り込んでも，もちろんその逆でもかまわない。

・SXIMAGE.Xのウィンドウはキャンバス.Xと同様，いくつでも開くことができる。

PIXファイルのアイコンをダブルクリックすればキャンバス.Xのウィンドウが自動的に開き，その中にダブルクリックしたアイコンの画像が表示される。これと同様のことをSXIMAGE.Xでも実現しようとすると，リソースを拡張しなくてはならない。そうすれば，CUTまたはPICファイルのアイコンをダブルクリックするだけでSXIMAGE.Xが起動するようになる。詳細は先月のSX信州.Xの解説を参照のこと。

アルゴリズム概説

ウィンドウシステムでは画像のデータ本体をG-RAMに置くことはできない。画面に表示されるのは，ほかのウィンドウなどに邪魔をされずに見える部分だけなのだから，画像本体はメインメモリに持っておき，必要な部分だけ表示するようにしなくてはならない。

1) PICファイルのヘッダを読み，表示する画像の大きさを調べる。

2) 画像の本体を格納するのに必要なだけの領域を確保する。たとえばフルサイズのPICだと，SX-WINDOW用(16色)に変換したあとで約90Kバイト必要である。もしそのウィンドウがいままで別の画像を表示していたのであれば，その画像のために確保してあった領域は放棄し，新たに確保しなおす。もしこの時点でメモリ確保に失敗すれば，SXIMAGE.Xは終了する。

3) 16色に変換する前のフルカラー画像(65536色)を一時的に展開する領域を確保する。もしメモリに余裕があるのなら，一気に画像の全体を展開したほうが処理が速いので，それだけの領域が取れるかどうか調べる。フルサイズなら512Kバイト，もっと小さな画像ならそれだけのサイズが確保できるかどうか調べる。

4) 確保できた場合→確保した領域に画像を一気に展開する。PIC.Rの展開ルーチンに若干の改造を加えたものを使っているので，展開のスピードは速い。展開が終わったら，6) 以降の縮小&16色変換に進む。

5) 確保できなかった場合→3ラスタ分だけの展開バッファを確保する。なぜ3ラスタかというのは，縮小ルーチンとのからみで

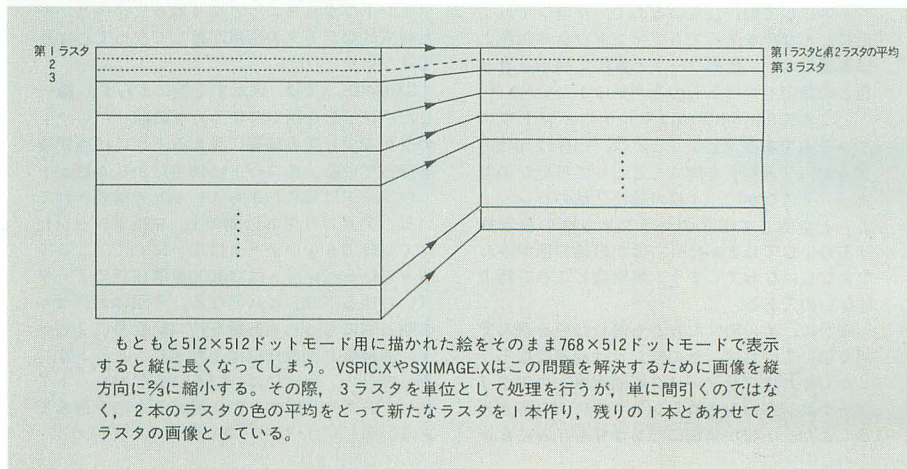
図1 画像表示プログラムの動作(その1)

＊1 g16形式……………SX-WINDOWのグラフィック形式のひとつ。16色ビットマップのなかではメモリ効率がよい。

＊2 768×512×65536……通常768×512ドットモードでは16色しか表示できないが，ビデオコントローラを直接操作すれば一応65536色も出せる。ただしグラフィックは512×512ドットの範囲しか正常に表示されない。VSPIC.Xでは残りの部分をマスクしている。

図2 (その2)

図3 縦2/3縮小によるドット比補正

もともと512×512ドットモード用に描かれた絵をそのまま768×512ドットモードで表示すると縦に長くなってしまう。VSPIC.XやSXIMAGE.Xはこの問題を解決するために画像を縦方向に2/3に縮小する。その際，3ラスタを単位として処理を行うが，単に間引くのではなく，2本のラスタの色の平均をとって新たなラスタを1本作り，残りの1本とあわせて2ラスタの画像としている。

ある。そうして確保した領域に，PICファイルから3ラスタ分ずつ画像を展開する。そしてそのたびごとに6)以降の縮小&16色変換を行う。3ラスタずつ展開する部分についてはかなり工夫を凝らした。この処理は多少複雑なうえにCで記述したので，速度の点で若干劣るが，その代わりメモリを食わなくてすむようになっている。

PICは，乱暴にいえば，「稲妻＋ペイント」で画像を生成するのだが，この稲妻は一瞬にして画面の下方まで走ってしまう。したがって，全画面分のバッファを取れる場合はいいのだが，3ラスタずつ展開するような場合は非常に困る。そこで，稲妻を一時的に蓄えておいて要求に応じて小出しにするようなバッファを作る。この稲妻バッファ(PIC.Rのアルゴリズムでは稲妻のことを連鎖と呼ぶので，正式には連鎖バッファ)は，カレントの3ラスタからはみ出した稲妻をどんどん溜め込む。新しい3ラスタの処理を始めるときに，稲妻バッファから3ラスタ分だけ稲妻を取り出す。新しい稲妻も，カレントの3ラスタをはみ出せば残りを稲妻バッファに蓄える。こうして，3ラスタずつの展開を実現している。稲妻バッファの実現には，双方向リストやガーベジコレクションなどの小技を用いている。土台となるPICの展開ルーチンは本誌1990年2月号のPIC.Rの記事のものを流用している。アセンブラで書き直す実力を持ったパワーユーザーの出現に期待しよう。

6)　画像を縦方向に縮小する。512×512ドットモード用に描かれた画像を768×512ドットモードで正しく表示するためには，ドットの縦横比を補正する必要がある。そのための処理。具体的には3ラスタを取ってきて，第1ラスタと第2ラスタの色を混ぜて1ラスタにする。これを第3ラスタと合わせれば2ラスタに縮小したことになる。この方式は，付録ディスクのVS2.Xから呼び出すVSPIC.Xでも使っている。VS.Xは768×512ドットモードで動作しているので，やはり縦方向の縮小をする必要があるのだ。

7)　65536色モードの画像を16色に変換する。これは1990年6月号の付録ディスクで載せたPIXファイル生成ツールsxconv.xと同じく，“桒野式アルゴリズム(たぶん誤差拡散法と同じものだと思う)”のカラー版を用いた。この階調落としのアルゴリズムは，簡単にいって，たとえば0.7の明るさの部分では平均して100個のうち70個のピクセルが点灯している状態を作るもので，その70個のピクセルができるだけ均等に，自然に散らばるようになっている。その処理をR，G，Bの3原色に対してそれぞれ行っている。

8)　16色に変換した画像は最初に確保した画像本体の領域に格納する。全画面分の変換が終了したなら，もはやフルカラー画像の領域は不要なので放棄する。そして初めて，グラフィックウィンドウを描画する。

*

以上より，SXIMAGE.XでPICファイルを表示させようとする際には，“プロセス情報”でメモリの空き領域がどのくらいあるか調べておくことが重要だ。最低でも150Kバイトくらいはないと画像が展開できない可能性がある(小さな絵ならもっと消費するメモリも少なくてすむ)。600Kバイト以上あると，PICファイルの展開に高速なものを使うようになるので多少は快適になる。しかし縮小&16色変換部をCで書いてしまっているので，この部分がネックになってしまい，全体としてはあまり速くない。

仕様上の不備

PICファイルとして現在サポートしているのは，最初に発表されたPIC.Rのフォーマットである32768色のPICファイルのみである。出回っているPICファイルのほとんどはこのタイプなのでほぼ問題なく使えるとは思うが，実のところ拡張はそれほど難しくない。つまりAPICに対応することも可能である。

SX-WINDOWはグラフィック画面768×512ドット，16色モードで動作しているが，これはPIC.Rの512×512ドット，65536色モードとはまったく整合しない。このギャップを吸収するために，前節で述べたようにいろいろと手段を講じている。最終的には，縦方向に縮小を加えさらに16色に変換した画像を表示している。つまり元の画像の情報が失われているのだ。これが気持ち悪いという人は気持ち悪いだろう。そこそこの画質が出ているので，この点については改めるつもりはない。

最後に

SXIMAGE.Xの原形はSXCUT.Xである。これは，今回のディスクに入っているSX信州やSXCLOCKの作者でもある吉川弘規氏の制作による。これを土台として，CUTファイル以外のファイルも表示できるように拡張したのがSXIMAGE.Xである。

したがって，SX-WINDOWのウィンドウまわりの処理は，ほぼ吉川氏のものを利用している。この場を借りてお礼を申し上げたい。

それから，なんといってもPIC.Rの作者である柳沢明氏にはいくら感謝してもし足りない。PICの優れた圧縮アルゴリズムはX68000のCGに革命を起こしたといってもいいだろう。

そしてSX-WINDOWのドキュメントには本当にお世話になった。限りなく嘘に近い記述があったおかげで丸々ひと晩費やす羽目になったこともあったが，あれなしにSXIMAGE.Xは完成しなかったことだろう。

統一パレット

SX-WINDOWはグラフィック16色の環境である。その16色は65536色のなかから任意に選ぶことができる。SX-WINDOW本体はテキスト画面をベースにして動作しているので，グラフィック用にどんな色を選んでもウィンドウ自体の色には影響しない。色のコードと実際に表示される色との対応をつけるものをパレットという。

ところでSX-WINDOWはマルチウィンドウのシステムである。ということは，一度に10枚の絵を飾って展覧会を開くことだって可能なのである。そのなかで，1枚の絵が「私のパレットよ」と主張して画面のグラフィックの色を独り占めにしてしまったら，ほかの絵の色がみんな台なしになってしまう。展覧会どころの話ではないのである。

確かに，その絵にもっとも適した色を選んで絵を描いてやれば，その絵だけは綺麗になる。しかしそれはグラフィック画面が単一のアプリケーションに占有されるシステムにのみ許されることだ。1枚が綺麗になるよりも，みんなが同時に見渡せることのほうが大切だ。それがウィンドウシステムというものである。資源の占有は許されない行為なのだ。そもそも，ほかのウィンドウからちょっと切り取って……といった将来的なデータの活用が難しくなってしまうではないか。

SXIMAGE.Xでは，表示する絵によらず，統一したパレットを用いている。その統一パレットのなかで少しでも綺麗に見えるように16色変換を行っている。色コードは16色だから4ビット

色コードは16色だから4ビットで構成されている。アルゴリズムの関係上，RGBプレーンだけで処理できないデータは扱いにくく，ここで使っているパレットはX68000標準16色のデータ(いわゆるRGBI)とは異なる。今回はピクセルの明るさにもっとも影響を持つ緑成分に2ビット，赤成分と青成分にはそれぞれ1ビットずつ割り振った。色合いは，いろいろなパレットで試してみて，比較的自然に見えるものを選んである。

参考文献
・柳沢　明，これが噂のPIC.R，Oh!X1990年2月号，pp.75-80
・SX-WINDOW開発資料，Oh!X1991年1月号

ダイアログで対話する(前編)

Nakamori Akira 中森　章

SX-WINDOWのユーザーインタフェイスで重要なポイントのひとつがダイアログウィンドウです。使い方がよくわからないと悩んでいた中森氏ですが，わかってしまえば意外に便利に使えるということです。さっそく解説をお願いしましょう。

前回は制御ボタンの基本的なところを学びました。今回はその応用としてスクロールバーを取り上げるつもりでした。が，やはり，スクロールバーの操作は現在の私の能力を超えていました（早い話がわからなかった)。そこで，今回は予定を変更してダイアログを取り上げることにします。

以前，私がSX-WINDOW上のライフゲームを作ったときは初期パターンとしてキーボードから入力されてくる文字列を利用しました。そのときは文字列を入力するためのウィンドウを実現するためにかなり長いプログラムを書く必要がありました。実はこのような処理はSX-WINDOWのマネージャのひとつであるダイアログマンの機能を使えば簡単に実現できてしまうのです。当時はダイアログの使い方が理解できなかったので，わざわざ長いプログラムを書いてしまったのですが，今となってはずいぶん無駄な行数を使ったものだと後悔しています。皆さんもこんな経験をしないように，ダイアログをしっかりと勉強しましょう。

ダイアログとは

ダイアログとは対話（dialog）を意味します。これは文字どおりプログラムとユーザーの対話を実現するための機能です。具体的には，

●なんらかの情報（エラーメッセージを含む）に対してユーザーに確認を促す場合

●プログラムの実行に必要なオプションの設定やファイル名などの付加情報をユーザーに入力させる場合

などに使用されます。図1にダイアログの例を示しましょう。図1(a)はSX-WINDOWの画面の右上にあるXのマーク上でポップアップメニューを出して「シェル情報」を選択したときに現れるウィンドウです。また図1(b)は同じく画面の右上のX68000のマークの上でポップアップメニューを出して「コントロールパネル」を選択したときに現れるウィンドウ内で，さらに「スイッチの設定」を選択したときに現れるウィンドウです。図1(a)は単にユーザーに確認を求めるためのダイアログ，図1(b)はもろもろの設定をユーザーに要求するためのダイアログになっています。

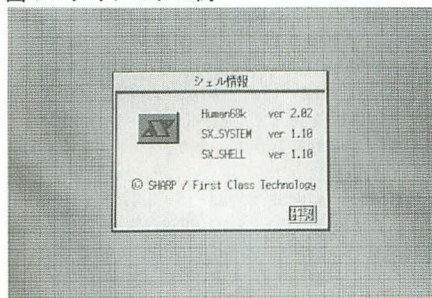
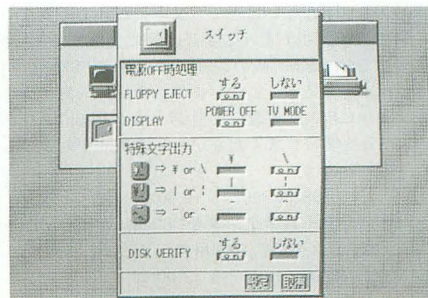
図1を見てわかるように，ダイアログとはひと言で制御ボタンが並んだウィンドウといえそうですね[1]。これが普通の制御ボタン付きのウィンドウと異なる点は，ダイアログが出ている間はタスクの切り替えが行われないという点です。すなわち，ダイアログが出ている状態ではユーザーが「確認」とか「取消」といったボタン（あるいはそれに相当するボタン）を押すまで，対話以外の処理は一切行われません（ほかのウィンドウの動作は停止しているのです)。これは，ダイアログがプログラムの実行に直接かかわる条件などを要求するものであるため，最優先で処理をする必要があるからなのでしょう。画面の動きがなくなってしまえば否がおうでも注意が引きつけられますからね。

ダイアログが，機能的には単に制御ボタンが並んだウィンドウではないといっても，実際は（プログラムするうえでは）制御ボタンの並んだウィンドウと理解しておけば十分だと思います。当然，ダイアログと同じ働きをするウィンドウを通常のウィンドウと制御ボタンの組み合わせでプログラムすることも可能です。また，このようにして実現したウィンドウも，それがユーザーとの対話を目的としたものである点で，ダイアログと呼ぶことができます。

ただし，通常のウィンドウと制御ボタンの組み合わせでダイアログを実現する場合は，前回説明したような，制御ボタンのオープンやクローズなどの煩わしい処理をプログラムで書かなければならず，結構厄介です。プログラムの行数にして100行程度はあっという間に増えてしまうでしょう。そして，このような煩わしさから私たちを解放してくれるのが今回説明しようとしているダイアログマンなのです。

ダイアログマンはダイアログウィンドウ内に配置しようとしている制御ボタンなどの情報をテーブルにしておけば，あとは(基本的には）3個の関数（ウィンドウのオープン，操作，クローズ）だけでダイアログ機能を実現してくれるありがたいマネージャなのです。

ところで，実行に必要な条件をいくつかの項目から選択するだけならポップアップメニューでも十分用が足せられるような気がします。そのような場合であってもダイアログを使うことが多いのは，やはりインパクトが違うからなのでしょう。ポップアップメニューでちまちまと項目を選択するより，画面の真ん中にドーンとウィンドウが出てきたほうが注意を引きやすいことはいうまでもありませんね。

図1　ダイアログの例

(a)　シェル情報

(b)　コントロールパネル（スイッチ）

1) 正確にはダイアログとは機能を示す言葉ですからウィンドウ自体を示しているわけではありませんが同一視しても差し支えはないでしょう。SX-WINDOWではダイアログに用いるウィンドウをダイアログウィンドウ(Macintoshではダイアログボックス)と呼んでいます。

ダイアログの操作

それでは，SX-WINDOWでダイアログを操作する手順について説明しましょう。ダイアログに関しても，これまでポップアップメニューや制御ボタンで経験してきたのと同様に，

ウィンドウのオープン
↓
操作
↓
ウィンドウのクローズ

が基本となります。そして，SX-WINDOWではそれぞれに対応する関数(システムコール)が用意されています。謹賀新年PRO-68K[2]（以下，おまけディスクと呼びます)のドキュメント(以下，単にドキュメントと呼びます)でダイアログマンの項を見てもらえば，それらが，

DMOpen
DMControl
DMClose(またはDMDispose)

であることがわかるでしょう。ただし，これらがわかっていてもダイアログ機能が使えるとは限りません(私もそのひとりでした)。ダイアログ機能を使いこなすためにはダイアログアイテムリストの理解が必要になります。これが，先に述べた制御ボタンの情報を格納しておくテーブルです。それでは，このダイアログアイテムリストとダイアログを扱う際の基本的な操作について順次説明していきましょう。

1) ダイアログアイテムリスト

ダイアログアイテムリストはダイアログアイテムのリスト(テーブル)です。ダイアログアイテムとは，ダイアログウィンドウ内に表示されるアイテム(その多くは制御ボタンです)のことで，具体的には，

ユーザーアイテム (DT_USER)
標準ボタン (DT_STDBTN)
セレクトボタン (DT_SELBTN)
オルタネートボタン (DT_OTNBTN)
スクロールバー等 (DT_RSCITM)
固定テキスト (DT_STCTXT)
編集可能テキスト (DT_EDTTXT)
アイコン (DT_ICNITM)
ピクチャー (DT_PICITM)

のどれかを指します。( )内はおまけディスクのsxlib.h(あるいはsxdef.h)というヘッダファイルをインクルードしたときに定義される名前(定数)です。ダイアログアイテムリスト内のアイテムの種類はこの名前で指定すればよいでしょう。

このうち，標準ボタン，セレクトボタン，オルタネートボタンは前回の制御ボタンで説明したものと同一です。固定テキストはウィンドウ上に書かれる文字列です。編集可能テキストはキーボードを使って入力される文字列です。これら以外のアイテムに関しては「リソース」と呼ばれるものに深くかかわっていて私にもまだよくわかりません。とにかく，これらの制御ボタンやテキストに関する位置，大きさ，種類などの情報を並べたものがダイアログアイテムなのです。そして，ダイアログアイテムの個数の情報と具体的なダイアログアイテムを並べたものがダイアログアイテムリストになります。ダイアログアイテムリストのイメージとしては，

個数の情報
ダイアログアイテム
ダイアログアイテム
ダイアログアイテム
ダイアログアイテム
:
:

というようになるでしょう。こころなしか，ポップアップメニューを定義するときに用意したメニューアイテムのデータ構造に似ていますね。

さて，ダイアログアイテムリストがおおよそどんなものであるかわかったところでC言語によるダイアログアイテムリストの表現方法を検討してみます。おまけディスクに付属したsxdef.hというヘッダファイル(これはsxlib.hの中でインクルードされている)の中では，ダイアログアイテムリストおよびダイアログアイテム(を示す構造体)の定義は，それぞれ，

```
typedef struct dlgIList {
    short  dlgILSize;
    short  dlgILData;
} dlgIList;
```

および，

```
typedef struct dlgItem {
    long            dlgIHdl;
    rect            dlgIBounds;
    unsigned char   dlgIType;
    unsigned char   dlgISize;
    unsigned short  dlgIData;
} dlgItem;
```

となっています。一応，構造体の各フィールドの意味を簡単に説明すると，

dlgILSize	ダイアログアイテムの個数−1
dlgILData	ダイアログアイテムの並び(?)
dlgIHdl	システムのワークエリア
dlgIBounds	アイテムを表示する位置
dlgIType	アイテムの種類
dlgISize	アイテムの初期値のサイズ
dlgIData	アイテムの初期値

となります。それぞれにどのような値を設定したらよいかの詳細はドキュメントでの説明に譲りますが，明らかにダイアログアイテムリストの定義(上側)は間違いですね。個数情報を表す，

short dlgILSize;

はいいとしても，それに続くダイアログア

今月のバグ出し

今月は気づいたバグはあまりありません(直しようのないバグはすでに本文で述べました)が，ひとつだけ大物があります。現在のままではダイアログウィンドウにひとつしか編集可能テキストがない場合でも文字列が入力できないことがあります(2度目以降に失敗する)。そこでSX-WINDOWのシステムであるFSX.Xにパッチを当てることにします。カレントディレクトリにFSX.Xを置いてリスト6のBASICプログラムを実行してください。リスト6が対象としているのは，

180998 90-06-15 12:00:00

というタイムスタンプを持つSX-WINDOWのver1.02のFSX.Xです。XVI用の，

248206 91-03-15 12:00:00

というタイムスタンプを持つSX-WINDOWのver1.10のFSX.Xではリスト6のfseekの引数を&HIFE91から&H2DA0Fに変更すればよいのですが，SX-WINDOWのver1.10ではテキストマンが変更されたせいかバグが発生しているようには見えません。それ以外のバージョンのFSX.Xではfseekの引数を&HIFE91から次のように変更してパッチを当てればいいのですが，ver1.02より古いバージョンを持っている人は早くバージョンアップを済ませておきましょう。

```
ver1.00    170052 90-03-15 12:00:00
           →&HIFE91を&HIE059に変更
ver1.01    180610 90-05-15 12:00:00
           →&HIFE91を&HIFD15に変更
```

イテムが,

```
        short    dlgILData;
```

という1行（たった2バイトのデータ）で表現できるはずがありません。結局，dlgIListというデータ型はプログラムでは使えそうにありません。

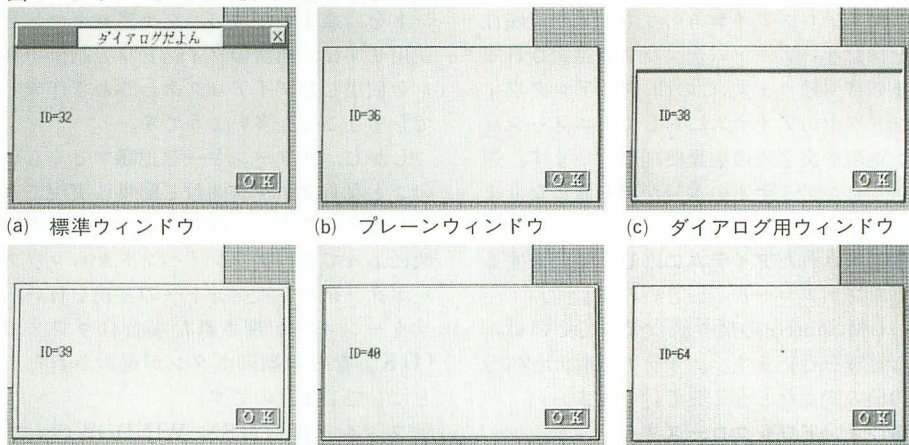
一方,ダイアログアイテムの定義(下側)のほうはそれなりに使えそうですが，最後のフィールド（dlgIData）のサイズがshort（2バイト）である点が問題です。このフィールドは制御ボタンのタイトル（表示される名前）やテキストの初期値を定義するフィールドなので，現実にはLASCII形式のデータがくることがほとんどです。LASCII形式のデータのサイズ(バイト数)は不定ですからとてもshort（2バイト）で収まるとは思えません。また,dlgIDataフィールドのサイズが不定になるからこそdlgISizeフィールドが意味を持ってくるのです。ここはdlgIDataフィールドのサイズを定義するフィールドです。ダイアログマンはdlgISizeフィールドのサイズを頼りにして，次のダイアログアイテムの先頭を認識するのです[3]。

このような理由から，C言語でダイアログアイテムを記述する場合は新たなdlgItemというデータ型を別に定義する必要がありそうです。それで，今回はdlgIDataのサイズを最大32バイトに固定した，

```
typedef struct dlgItem2 {
    long            dlgIHdl;
    rect            dlgIBounds;
    unsigned char   dlgIType;
    unsigned char   dlgISize;
    unsigned char   dlgIData32;
} dlgItem2;
```

というデータ型を定義することにします。ただし，このデータ型を使用する場合は，dlgISizeフィールドの値を必ず32にしなければなりません。このときダイアログアイテムリストは，

```
struct {
    short    dlgILSize;
    dlgItem2 dItem1;
    dlgItem2 dItem2;
    dlgItem2 dItem3;
    dlgItem2 dItem4;
        :
        :
} dItemList;
```

図2　ダイアログとして使えそうなウィンドウ

(a)　標準ウィンドウ　(b)　プレーンウィンドウ　(c)　ダイアログ用ウィンドウ
(d)　エラーダイアログ用ウィンドウ　(e)　漢字変換用ウィンドウ　(f)　電卓用ウィンドウ

などという構造体で表すことができるようになります。このあとはC言語の構造体の初期化の方法に従ってダイアログアイテムリストの内容を記述するだけです。

例として標準ボタンと固定テキストをひとつずつ持つダイアログのアイテムリストの定義を以下に示します。ダイアログアイテムリストがどういうものであるか実感してくださいね。

```
struct {
    short    itemNo;
    dlgItem2 dItem1;
    dlgItem2 dItem2;
} dItemList ={
        2−1,
    {
        0,
        {206,102,248,120},
        DT_STDBTN,
        32,
        "¥007 O K "
    },
    {
        0,
        {16,50,240,62},
        DT_STCTXT+DT_DISABL,
        32,
        "¥001¥020ダイアログなのだ"
    }
};
```

なお，この手のアイテムリストはポップアップメニューで「…について」などと書かれた項目を選択したときに現れるダイアログでよく使用されます。

ところで，DT_DISABLは未帰還属性を示すシンボルで，sxlib.h（またはsxdef.h）というヘッダファイルをインクルードしたときに定義されます。これは，ダイアログを制御するDMControlという関数（あとで説明します）からの戻りを規定する定数です。未帰還属性が付属したアイテムに関しては，それがマウスで選択されてもDMControlから戻りません。

2)　基本的な操作

ダイアログマンを使ってダイアログを実現する場合の手順と必要な関数を以下に説明します。用意したダイアログアイテムリストは新たに確保した領域にコピーして使用しなければならないことに気づけば，処理自体は簡単です。

●領域を確保する

システムコール：MMChHdlNew

MMChHdlNewによってダイアログアイテムリストを格納する領域を確保し，その領域へのハンドルを獲得します。このハンドルは後ろのDMOpenで参照されます。

●ダイアログアイテムリストをコピーする

ライブラリ関数：memcpy

MMChHdlNewで確保した領域にダイアログアイテムリストの内容をコピーします。これで，やっとダイアログアイテムリストが使える状態になります。

●ウィンドウをオープンする

システムコール：DMOpen

ダイアログウィンドウをオープンします。引数は，先に獲得したダイアログアイテムリストへのハンドルが加わるほかは，通常のウィンドウをオープンするWMOpenへの引数と同じです。なお，WMOpen(やDMOpen)への第5引数であるウィンドウのリソースID（ウィンドウの種類）は標準では9種類が用意されています。このうち，ダイアログウィンドウとして使えそうなのはIDが32，36，38，39，40，64であるウィンドウでしょう（図2）。

●ボタンが押されるのを待つ

システムコール：DMControl

ダイアログアイテムのうち，未帰還属性を持たないアイテムがマウスで選択されるまで待ち続けます。この間，ダイアログウィンドウ上のアイテムに対してアニメーション処理や文字列の編集処理を行います。選択されたアイテムの番号が戻り値になります。

●選択されたアイテムに応じた処理をする

システムコール：いろいろ

DMControlの戻り値に従っていろいろな処理を行います。アイテム(制御ボタン)の値の変更などもここで行います。

●ウィンドウをクローズする

システムコール：DMCloseあるいはDMDispose

用がすんだらダイアログウィンドウをクローズします。DMOpenの第1引数が0でない場合（領域を指定する）はDMCloseを使用し，第1引数が0の場合（メモリマンに領域の確保を任せる）はDMDisposeを使用します。

●領域を解放する

システムコール：MMHdlDispose

最後に，ダイアログアイテムの領域を解放します。

---

2) 本誌1991年1月号の付録ディスク。この連載は付録ディスクのドキュメントを読解する方法を示すことが建て前になっている(知ってました？)。
3) このとき，各ダイアログアイテムがメモリの偶数番地から配置されるようにdlglData，dlglSizeフィールドの値を調整する必要がある。構造体dlgltemはdlglDataフィールド以外のサイズは14バイト（偶数）であるから，dlglDataフィールドのサイズも偶数バイトにすること（奇数バイトになるときはダミーなデータを詰める)。

フィルタ関数

これまでで，ダイアログマンを使ってダイアログを実現する方法の説明がひと通り終わりましたが，プログラム例に入る前に，DMControlの引数であるフィルタ関数について説明しておきましょう[4]。これは，初心者にはとっつきにくいかもしれませんが，慣れてしまうとなかなか便利な関数です。

フィルタ関数の最大の目的は発生したイベントを加工することです。たとえば，ダイアログ内の「ＯＫ」とか「確認」といったボタンをマウスで選択する代わりにキーボードのリターンキーを押してもよいことにしたい場合があります。リターンキーを押すとキーダウンイベントが発生しますが，通常このイベントはダイアログでは無視されてしまいます。そこで，キーダウンイベントを認識するために，ダイアログマンを使用せずに，通常のウィンドウと制御ボタンを使用したダイアログをわざわざ作成してしまうことも多いようです。

しかし，リターンキーを認識するくらいのことならフィルタ関数で簡単に実現できてしまうのです。具体的には，フィルタ関数によってキーダウンイベントからマウス左ボタンダウンイベントへの変換を行い，リターンキーが押された場合はマウスで「ＯＫ」などの制御ボタンが選択されたことにしてしまうのです。

フィルタ関数はSX-WINDOWのシステムによって，

第1引数：ダイアログへのポインタ
第2引数：イベントレコードへの
　　　　　ポインタ

という引数が積まれてコールされます。そこで，フィルタ関数内でこの第2引数を解析し，自分の都合のよいようにイベントを書き換えてやるのです。つまり，第2引数のeWhatフィールドがE_KEYDOWNであれば，

eWhatフィールド　←　E_MSLDOWN
eWhereフィールド　←　ボタン内のどこか
（グローバル座標）

というようにイベントを書き換えれば[5]リターンキーを押すことが制御ボタンを選択することと同じことにできるのです。このように，フィルタ関数にはいろいろと面白い使い方がありそうですね。

---

4) フィルタ関数が不要な場合（この場合のほうが多い）は，DMControlの引数には0を指定する。
5) E_KEYDOWN（キーダウンイベント），E_MSLDOWN（マウス左ボタンダウンイベント）はsxlib.h(sxdef.h)内で定義されている定数。eWhat，eWhereは同じくsxlib.h(sxdef.h)内で定義されているeventという構造体のフィールド名。

サンプルプログラム

それでは，ダイアログマンを使ったサンプルプログラムを紹介します。今回のプログラムは汎用性を持たせるため，ダイアログを行う部分を関数（doDialogという名前）にしてあります。リスト1のスケルトン（骨格）プログラムとリンクして使ってください。ポップアップメニューで「ダイアログを開く」を選択すると，ダイアログウィンドウが開くようになっています。なお，リスト1と前回（1991年4月号）で示したスケルトンプログラムの違いは，ポップアップメニューのアイテムリストの定義（MNITEMSとMNILIST）とマウス右ボタンダウンイベントの処理（procMSRDOWN関数）だけですから，前回のスケルトンプログラムを打ち込んでいる人はそこを変えるだけで結構です。

さて，スケルトンプログラムとリスト3以降に示すダイアログプログラム（関数）をリンクする方法について説明しておきましょう。リスト2は前回も示したコンパイル用バッチファイルです。リスト2の(a)～(c)のどれか1行を書き込んだバッチファイルの名前を，

SXCC.BAT

とし，リスト1のスケルトンプログラムが，

SX_SKEL.C

という名前，リスト3以降のダイアログプログラムが，

SX_DLOG.C

という名前であるとします。このとき，

SXCC SX_SKEL.C SX_DLOG.C

というコマンドを実行すると，2つのプログラムのリンクが行われ，

SX_SKEL.X

という実行ファイルが作られます。コンパイルに関する動作環境などは前回の連載を参照してください。

ところで，リスト3以降のダイアログはリスト1のスケルトンプログラムとは独立な関数として書かれていますから，各自のオリジナルなスケルトンプログラムから呼び出して使用することもできます。試してみてくださいね。

1) 単純なダイアログ

まずは小手調べです。最も単純なダイアログを作ってみます。リスト3は確認をするための制御ボタン（ＯＫと表示される）がひとつと，3つの文字列からなるダイアログを開くプログラムです。このようなダイアログはSX-WINDOW上のアプリケーションでプログラムのバージョンや作成者を表示するために使われることが多いようです。リスト3で行っているダイアログの操作は先に示した手順とまったく同一ですから復習のつもりでプログラムを読んでみましょう。

なお，リスト3ではフィルタ関数（MyFiletr）を使用してリターンキーのキーダウンイベントをマウス左ボタンダウンイベントに変換しています。特定の制御ボタンが選択されたように見せかけるためにイベントレコードのeWhereフィールドを書き換えていますが，ここに設定するマウスの座標はグローバル座標であるということに注意しましょう。ダイアログアイテムリス

ト内に指定されている制御ボタンなどの位置はローカル座標なのでそれらをグローバル座標に変換して考える必要があります。

また，ダイアログに関しては，DMOpenの引数であるダイアログウィンドウをオープンする位置（グローバル座標）の決定も結構悩みの種です。SX-WINDOWの画面が768×512ドットであることを考慮して，その中心である（384,128）という座標を囲むようにウィンドウをオープンすればよいでしょう。現実には画面の中心よりやや上部にウィンドウをオープンすればバランスのよいダイアログになります。

2）制御ボタンやテキストを持つダイアログ

リスト3のようにただ文字を表示するだけではダイアログ（対話）という観点からは面白くありません。もっとダイアログらしいプログラムを作ってみます。リスト4のプログラムではダイアログウィンドウ内に，

・OK（確認）ボタン
・取消ボタン
・オルタネートボタン
・セレクトボタン
・編集可能テキスト

を持っています[6]。このくらい賑やかであれば対話するという気がするでしょう。

リスト4のプログラムについて簡単に解説しましょう。リスト4のウィンドウではオルタネートボタン，セレクトボタン，編集可能テキストが値（や文字列）を持っていますから，それらの値に対する考慮をしなければなりません。すなわち，いったん設定した値は覚えておいて，次にダイアログを開いたときに初期値として表示してやる程度の芸当は必要でしょう。このためには，ダイアログマンのあと2～3の関数およびコントロールマン，テキストマンのいくつかの関数の助けを借りることになります。

前回を思い出してください。制御ボタンに値を設定したり，制御ボタンの値を参照する関数に，

CMValueSet
CMValueGet

がありました。ダイアログウィンドウ上に表示されているアイテムとはいえ，実体は制御ボタンには変わりありませんから，値の参照や設定のためにはこれらの関数を実行することになります。ただし，そのためには制御ボタンへのハンドルが必要です。ところが，ダイアログマンを使用してダイアログウィンドウをオープンする場合は，制御ボタンのオープンなどはすべてダイアログマンがやってくれるので，そのままでは制御ボタンへのハンドルを知ることができません。そこで，制御ボタンへのハンドルを知る関数を利用します。それは，

DIGet

という関数です。これは，ダイアログアイテムリストのアイテム番号を指定して，それに対応する制御ボタンのハンドル，種類，位置を教えてもらうための関数です。この関数でハンドルを得てしまえば，制御ボタンの値を自由に操作することができるようになりますね。制御ボタンの値を変更したあとは，

CMDrawOne

関数で，こまめに制御ボタンを書き直す必要があります。もし，ダイアログウィンドウ上のすべてのアイテムを一度に書き直すのであれば，

DMDraw

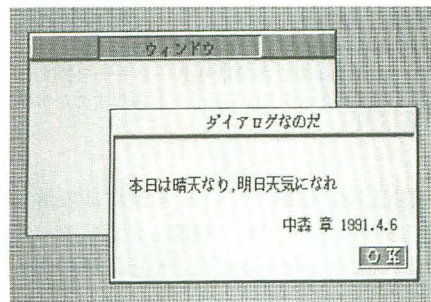
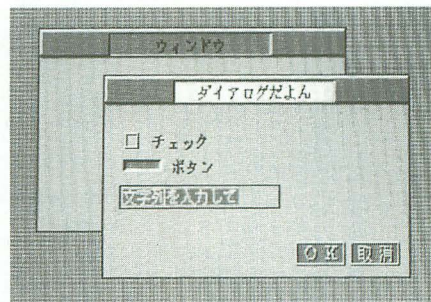
という関数ひとつで十分です。

制御ボタンの値を変更するためには，それがマウスで選択されたときにDMControlから呼び出した側に制御がいったん戻る必要があります[7]。ただし，ダイアログウィンドウ自身は「OK」ボタン，「取消」ボタンを押すまでは開いたままにしておきたいですから，制御ボタンの値を変更して書き直しを行ったあとは再びDMControlを呼び出すようにしています。ここらへんの制御は「OK」または「取消」ボタンが押されるまでの無限ループで実現しているのがわかるでしょう。

なお，ダイアログウィンドウ上のテキスト（固定テキスト，変数可能テキスト）の文字列を操作するためには制御ボタンとは別の関数が必要です。

DITGet　　（文字列の獲得）
DITSet　　（文字列の設定）

がその関数です。これらはDIGetで得たハンドルをもとに文字列を操作します。このときのハンドルはテキストエディット（tEdit型）ではなく文字列そのものへのハンドルとなっています。したがって，このハンドルではテキストマンで提供されている関数を呼ぶことができませんので注意しましょう。

ところで，ひとつのダイアログウィンドウ上に編集可能文字列を複数持つことも（理論上は）可能です。ただし，現在のSX-WINDOW（ver1.02，ver1.10とも）のシステムでは，2つ以上の編集可能文字列があるとDITGet関数で入力した文字列が正しく取り出せないようです。多くの場合，一番最後に入力した文字列しか入力されたことになりませんでした（バグかな）。また，タブを入力することで編集対象の文字列を切り替えられるようになっているみたいですが，実際にタブを入力すると暴走してしまいます。この暴走をくい止めるため，リスト4ではフィルタ関数で入力されたタブを空白に変換するようにしています。さらに，最新版（ver1.10）のSX-WINDOWではDMOpenだけでは編集可能文字列に正しく初期値が設定できない（グレードダウン!?）ようなので，初期値が必要な場合はダイアログウィンドウをオープンしたあとに必ずDITSet関数で文字列の初期値を設定しなければなりません[8]。

長々と解説してきましたが，リスト4のプログラムを理解できれば，よほど複雑なダイアログでない限りは自由に作れるようになるでしょう。リスト4のプログラムではセレクトボタンはひとつしか表示しませんが，2つ以上なくて何がセレクト（複数からひとつを選択）するためのボタンだ，という意見もあると思います。基本はこれまでの説明で足りていると思いますからどんどん改造してみてくださいね。

蛇足ですが，リスト4のフィルタ関数では，奇をてらってリターンキーの入力を「取消」ボタンの選択と等しいことにしています[9]。単なる意地悪です。ごめんなさい。

リスト3の実行例

リスト4の実行結果

6）Macintoshの世界ではダイアログアイテムの1番目が確認ボタン，2番目が取消ボタンであることは常識らしい。
7）当然，値を変更する可能性のあるアイテムに未帰還属性をつけてはいけません。
8）SX-WINDOWのver1.10は専用エディタが付属

するなどしてテキストエディット機能が強化された半面，従来のテキストエディット機能と互換性のない部分が出てきたように思う。
9)　Macintoshの世界ではリターンキーを押すことは確認ボタンを選択したこととみなすのが常識らしい。

＊

SXLIFEを作っているときには理解できなかったダイアログ機能も最近になってわかるようになりました。やっと胸の支えが取れたような気分です。わかってみるとダイアログ機能は案外簡単で便利な機能なのですね。次回はダイアログ機能の続きとして，リソースを使ったダイアログとか，ダイアログマンを使わないダイアログなんかを説明してみたいと思っています。

ところで，この連載の各回のテーマは私の趣味で行き当たりばったり的に取り上げているのですが，ぜひ取り上げてもらいたいテーマがあれば編集部までお便りをください。できるだけ読者の要望に沿ったテーマを取り上げていきたいと思います。それでは次回まで……。

《参考文献》
1)　中森　章，「SXLIFE Part IIポップアップメニューの追加」，Oh！X 1991年2月号，pp.116-119.
2)　中森　章，「SXLIFE Part IIIライフゲームで姓名判断？」，Oh！X 1991年3月号，pp.104-113.

リスト1　スケルトンプログラム

```
  1: /*
  2:
  3:      S X - W I N D O Wスケルトンプログラム
  4:
  5:                 (C) 中森　章, Feb. 3, 1991
  6: */
  7: #include <stdio.h>
  8: #define __POINT_T          /* point_t 型を使う */
  9: #include <sxlib.h>
 10: #define FALSE   0
 11: #define TRUE    ~FALSE
 12: /*
 13:           ここでウィンドウに関する定数を設定
 14: */
 15: #define WDEFID           WI_STD
 16: #define WINOPT           ( WC_GBOX | WC_GBOXON )
 17: #define WINWIDTH         0x100
 18: #define WINHIGHT         0x080
 19: #define WINTITLE         "¥012ウィンドウ"
 20: #define EVENTMASK        EM_EVERY
 21:
 22: #define MDEFID           1
 23: #define MNENABLE         0xffffffff
 24: #define MNITEMS          1
 25: #define MNILIST          "¥0¥0¥021ダイアログを出す "
 26: #define MNTITLE          "¥014メニューだよ"
 27: /*
 28:           ここは定数から計算される定数
 29: */
 30: #define WINOPTL          ( WINOPT & 0xf )
 31: #define WINDEFID         ( WDEFID << 4 | WINOPTL )
 32:
 33: window  *winPtr;
 34: rect    winSize;
 35: event   eventRec;
 36: int     activeFlag;
 37:
 38: int     ctrlFlag;
 39: int     menuFlag;
 40:
 41: menu    **menuHdl;
 42:
 43: menu theMenu = {
 44:     0,0,0,0,MNENABLE,0,{MNITEMS-1,MNILIST}
 45: };
 46:
 47: main()
 48: {
 49:     if( SX_init()==FALSE ) OpenError() ;
 50:     while( 1 ){
 51:         TSEventAvail(EVENTMASK,&eventRec);
 52:         switch( eventRec.eWhat ){
 53:         case E_IDLE:     procIDLE();     break;
 54:         case E_MSLDOWN:  procMSLDOWN();  break;
 55:         case E_MSLUP:    procMSLUP();    break;
 56:         case E_MSRDOWN:  procMSRDOWN();  break;
 57:         case E_MSRUP:    procMSRUP();    break;
 58:         case E_KEYDOWN:  procKEYDOWN();  break;
 59:         case E_KEYUP:    procKEYUP();    break;
 60:         case E_UPDATE:   procUPDATE();   break;
 61:         case E_ACTIVATE: procACTIVATE();break;
 62:         case E_SYSTEM1:  procSYSTEM();   break;
 63:         case E_SYSTEM2:  procSYSTEM();   break;
 64:         case E_USER1:    procUSER();     break;
 65:         case E_USER2:    procUSER();     break;
 66:         }
 67:     }
 68: }
 69:
 70:
 71: SX_init()
 72: {
 73:     task    taskBuf;
 74:
 75:     TSGetTdb(&taskBuf, -1);
 76:     if( (TSTakeParam(&taskBuf.command,&winSize,NULL,0,NULL,NU
LL)&1)==0 ){
 77:         *(int *)&winSize.left = TSGetWindowPos();
 78:         winSize.right = winSize.left+WINWIDTH;
 79:         winSize.bottom= winSize.top +WINHIGHT;
 80:     }
 81:     winPtr=WMOpen(NULL,&winSize,WINTITLE,TRUE,WINDEFID,(windo
w *)-1,TRUE,TSGetID());
 82:     if( winPtr == NULL ) return( FALSE );
 83:     winPtr->wOption = WINOPT;
 84:     activeFlag=FALSE;
 85:     ctrlFlag  = CtrlPrepare();/* コントロールが不要なら ctrlF
lag=FALSE */
 86:     menuFlag  = MenuPrepare();/* メニューが不要なら     menuF
lag=FALSE */
 87:     drawGrowBox();
 88:     return( TRUE );
 89: }
 90:
 91: SX_term()
 92: {
 93:     if( ctrlFlag ) CtrlDispose();
 94:     if( menuFlag ) MenuDispose();
 95:     WMDispose( winPtr );
 96:     exit();
 97: }
 98:
 99: drawGrowBox()
100: {
101:     GMSetGraph( winPtr );
102:     WMDrawGBox( winPtr );
103: }
104:
105: CtrlPrepare()
106: {
107:     return( FALSE );
108: }
109:
110: CtrlDispose()
111: {
112:     return( FALSE );
113: }
114:
115: MenuPrepare()
116: {
117:     menuHdl=(menu**)MMChHdlNew( sizeof(theMenu) );
118:     if( menuHdl == NULL ) return ( FALSE );
119:     memcpy(*menuHdl,&theMenu,sizeof(theMenu));
120:     (*menuHdl)->mProc=RMRscGet(('M'<<24)|('D'<<16)|('E'<<8)|'
F',MDEFID);
121:     if( (int)((*menuHdl)->mProc)<=0 ){
122:         MMHdlDispose(menuHdl);
123:         return( FALSE );
124:     }
125: #if MDEFID==1
126:     (*menuHdl)->mHandle=MNTITLE;
127: #endif
128:     return( TRUE );
129: }
130:
131: MenuDispose()
132: {
133:     MMHdlDispose(menuHdl);
134:     return( TRUE );
135: }
136:
137: procIDLE()
138: {
139:     return( FALSE );
140: }
141:
142: procMSLDOWN()
143: {
144:     if( eventRec.eWhom != winPtr ) return( FALSE );
145:     if( activeFlag == FALSE ){
146:         WMSelect( winPtr );
147:         activeFlag = TRUE;
148:         if( EMLStill() == 0){
149:             TSGetEvent(EVENTMASK,&eventRec);
150:             return( FALSE );
151:         }
152:     }
153:     switch( SXCallWindM(winPtr,&eventRec) ){
154:     case W_INCLOSE:
155:         SX_term(); break;
156:     case W_INGROW:
157:     case W_INZMOUT:
158:     case W_INZMIN:
159:         GMClipRect(&winPtr->wGraph.grRect);
160:         break;
161:     }
162:     TSGetEvent(EVENTMASK,&eventRec);
163:     return( TRUE );
164: }
165:
166: procMSLUP()
167: {
168:     return( FALSE );
169: }
170:
171: procMSRDOWN()
172: {
173:     int item;
174:     char BUF[128];
175:
176:     if( eventRec.eWhom != winPtr ) return( FALSE );
177:     GMSetGraph( winPtr );
178:     if( activeFlag == FALSE ){
179:         WMSelect( winPtr );
180:         activeFlag = TRUE;
181:         if( EMRStill() == 0){
182:             TSGetEvent(EVENTMASK,&eventRec);
183:             return( FALSE );
184:         }
185:     }
```

```
186:     item=MNSelect(menuHdl,eventRec.eWhere);
187:     TSGetEvent(EVENTMASK,&eventRec);
188:     if(item!=0){
189:         doDialog();
190:     }
191:     return( TRUE );
192: }
193:
194: procMSRUP()
195: {
196:     return( FALSE );
197: }
198:
199: procKEYDOWN()
200: {
201:     return( FALSE );
202: }
203:
204: procKEYUP()
205: {
206:     return( FALSE );
207: }
208:
209: procUPDATE()
210: {
211:     if( eventRec.eWhom != winPtr ) return( FALSE );
212:     WMUpdate( winPtr );
213:     if( ctrlFlag ) CMDraw( winPtr );
214:     WMUpdtOver( winPtr );
215:     drawGrowBox();
216:     TSGetEvent(EVENTMASK,&eventRec);
217: }
218:
219: procACTIVATE()
220: {
221:     if( eventRec.eWhom == winPtr )   activeFlag = TRUE;
222:     else if( eventRec.eWhom != NULL ){
223:         if( activeFlag ) {
224:             activeFlag = FALSE;
225:             TSGetEvent(EVENTMASK,&eventRec);
226:         }
227:     }
228:     return( TRUE );
229: }
230:
231: procSYSTEM()
232: {
233:     switch( ((tsevent*)&eventRec)->what2 ){
234:     case CLOSEALL:
235:     case ENDTSK:
236:         SX_term(); break;
237:     case WINDOWSELECT:
238:         WMSelect( winPtr ); break;
239:     }
240: }
241:
242: procUSER()
243: {
244:     return( FALSE );
245: }
246:
247: OpenError()
248: {
249:     DMError(0x101,"ウィンドウがオープンできません");
250:     SX_term();
251: }
```

リスト2　コンパイル用バッチファイル

```
a) XC Ver.1.0用

cc /s4k /h8k  %1 %2 %3 %4 %5 %lib%¥sxlib.a


b) XC Ver.2.0用

cc /Gs4k /Gh8k %1 %2 %3 %4 %5 %lib%¥__mainc.o %lib%¥sxlib.a


c) GCC用

gcc -O  %1 %2 %3 %4 %5 %lib%¥sxlib.a
```

リスト3　単純なダイアログ

```
  1: /**************************************
  2:  *  ダイアログのサンプルプログラム  *
  3:  *                                  *
  4:  *           1991.4.6     中森  章   *
  5:  **************************************/
  6: #include <stdio.h>
  7: #define __POINT_T        /* point_t 型を使う */
  8: #include <sxlib.h>
  9: #define FALSE   0
 10: #define TRUE    ~FALSE
 11: /*
 12:     ここでダイアログウィンドウに関する定数を設定
 13: */
 14: #define DWINDEFID       (38<<4)
 15: #define DWINTITLE       "¥020ダイアログだよん"
 16: /*
 17:     アイテムリスト(sxlib.h内の定義だけで書くのはキツイ)
 18: */
 19: typedef struct dlgItem2 {
 20:     long            dlgIHdl;
 21:     rect            dlgIBounds;
 22:     unsigned char   dlgIType;
 23:     unsigned char   dlgISize;
 24:     unsigned char   dlgIData[32]; /* 初期値データサイズを */
 25:                         /* 32バイトに固定した型 */
 26: } dlgItem2;
 27:
 28: struct {
 29:     short     itemNo;
 30:     dlgItem2 dItem1;
 31:     dlgItem2 dItem2;
 32:     dlgItem2 dItem3;
 33:     dlgItem2 dItem4;
 34: } dItemList ={
 35:     4-1,            /* ダイアログの個数－1 */
 36:     {
 37:     0,              /* ハンドル等        */
 38:     {256-8-42,128-8-18,256-8,128-8},    /* 境界 */
 39:     DT_STDBTN,      /* 標準ボタン        */
 40:     32,             /* 初期値データサイズ! */
 41:     "¥007 O K "   /* 初期値データ      */
 42:     },
 43:     {
 44:     0,              /* ハンドル等        */
 45:     {0,4,256,16},   /* 境界         */
 46:     DT_STCTXT+DT_DISABL,  /* 固定テキスト+未帰還属性 */
 47:     32,             /* 初期値データサイズ   */
 48:     "¥001¥020ダイアログなのだ" /* 初期値データ,中央寄せ */
 49:     },
 50:     {
 51:     0,              /* ハンドル等        */
 52:     {16,50,240,62},     /* 境界         */
 53:     DT_STCTXT+DT_DISABL,     /* 固定テキスト+未帰還属性 */
 54:     32,             /* 初期値データサイズ   */
 55:     "¥000¥035本日は晴天なり,明日天気になれ" /* 初期値データ,
左寄せ */
 56:     },
 57:     {
 58:     0,              /* ハンドル等        */
 59:     {240-96,80,240,92},  /* 境界         */
 60:     DT_STCTXT+DT_DISABL,  /* 固定テキスト+未帰還属性 */
 61:     32,             /* 初期値データサイズ   */
 62:     "¥377¥020中森 章 1991.4.6" /* 初期値データ,右寄せ */
 63:     }
 64: };
 65:
 66: /*
 67:     ダイアログを開く位置(中央よりも少し上にしてある)
 68: */
 69: rect dlBounds={ 384-128,256-64-20,384+128,256+64-20 };
 70:
 71: /*
 72:     フィルタ関数
 73: */
 74: MyFilter(Dialog,ev)
 75: dialog *Dialog;
 76: event  *ev;
 77: {
 78:     point_t okbtn;
 79:
 80:     if( ev->eWhat == E_KEYDOWN ){
 81:         if((short)(ev->eWhom)==13){
 82:             okbtn.p.x=384+128-10;
 83:             okbtn.p.y=256+64-20-10;
 84:             ev->eWhere=okbtn;
 85:             ev->eWhat =E_MSLDOWN;
 86:         }
 87:     }
 88:     return 0;
 89: }
 90:
 91:
 92: /*********************************************
 93:     ダイアログを使う
 94: *********************************************/
 95: doDialog()
 96: {
 97:     dialog   *dialogPtr;
 98:     dlgIList **dIHdl;
 99:     int       ditem;
100:
101: /* (1) 領域を確保する */
102:
103:     dIHdl=(dialog**)MMChHdlNew( sizeof(dItemList) );
104:     if( dIHdl == NULL ){
105:         DMError(0x101,"領域確保に失敗しました。");
106:         return ( FALSE );
107:     }
108:
109: /* (2) アイテムリストをコピーする */
110:
111:     memcpy(*dIHdl,&dItemList,sizeof(dItemList));
112:
113: /* (3) ウインドウをオープンする */
114:
115:     dialogPtr=DMOpen(NULL,&dlBounds,DWINTITLE,TRUE,DWINDEFID,

116:             (window *)-1,TRUE,TSGetID(),dIHdl);
117:     if( dialogPtr == NULL ){
118:         MMHdlDispose(dIHdl);
119:         DMError(0x101,"ウインドウがオープンできません。");
120:         return( FALSE );
121:     }
122:
123: /* (4) ボタンが押されるのを待つ */
124:
125:     ditem=DMControl((void*)MyFilter );
126:
127: /* (5) 選択されたアイテムに応じた処理をする */
128:
129:
130: /* (6) ウインドウをクローズする */
131:
132:     DMDispose(dialogPtr);
133:
134: /* (7) 領域を解放する */
135:
136:     MMHdlDispose(dIHdl);
137:
138: };
```

リスト4　より一般的なダイアログ

```
  1: /****************************************
  2:  *  ダイアログのサンプルプログラム  *
  3:  *                                *
  4:  *          1991.4.6     中森 章   *
  5:  ***************************************/
  6: #include <stdio.h>
  7: #define __POINT_T        /* point_t 型を使う */
  8: #include <sxlib.h>
  9: #define FALSE   0
 10: #define TRUE    ~FALSE
 11: /*
 12:         ここでダイアログウィンドウに関する定数を設定
 13: */
 14: #define DWINDEFID        (WI_STD<<4)
 15: #define DWINTITLE        "¥020ダイアログだよん"
 16: /*
 17:   アイテムリスト(sxlib.h内の定義だけで書くのはキツイ)
 18: */
 19: typedef struct dlgItem2 {
 20:     long            dlgIHdl;
 21:     rect            dlgIBounds;
 22:     unsigned char   dlgIType;
 23:     unsigned char   dlgISize;
 24:     unsigned char   dlgIData[32]; /* 初期値データサイズを */
 25:                                   /* 32バイトに固定した型 */
 26: } dlgItem2;
 27: 
 28: struct {
 29:     short    itemNo;
 30:     dlgItem2 dItem1;
 31:     dlgItem2 dItem2;
 32:     dlgItem2 dItem3;
 33:     dlgItem  dItem4;
 34:     dlgItem2 dItem5;
 35:     dlgItem2 dItem6;
 36: } dItemList ={
 37:         6-1,                    /* ダイアログの個数-1 */
 38:     {
 39:         0,                      /* ハンドル等       */
 40:         {256-8-46-42,128-8-18,256-8-46,128-8},  /* 境界 */
 41:         DT_STDBTN,              /* 標準ボタン       */
 42:         32,                     /* 初期値データサイズ! */
 43:         "¥007 O K "             /* 初期値データ     */
 44:     },
 45:     {
 46:         0,                      /* ハンドル等       */
 47:         {256-8-42,128-8-18,256-8,128-8},     /* 境界 */
 48:         DT_STDBTN,              /* 標準ボタン       */
 49:         32,                     /* 初期値データサイズ! */
 50:         "¥007 取 消 "           /* 初期値データ     */
 51:     },
 52:     {
 53:         0,                      /* ハンドル等       */
 54:         {16,16,16+24+48,34},    /* 境界             */
 55:         DT_OTNBTN,              /* オルタネートボタン */
 56:         32,                     /* 初期値データサイズ! */
 57:         "¥010チェック"          /* 初期値データ     */
 58:     },
 59:     {
 60:         0,                      /* ハンドル等       */
 61:         {16,40,16+30,48},       /* 境界             */
 62:         DT_SELBTN,              /* セレクトボタン   */
 63:         2,                      /* 初期値データサイズ */
 64:         0                       /* 初期値データ     */
 65:     },
 66:     {
 67:         0,                      /* ハンドル等       */
 68:         {50,40,50+48,52},       /* 境界             */
 69:         DT_STCTXT+DT_DISABL,    /* 固定テキスト+未帰還属性 */
 70:         32,                     /* 初期値データサイズ */
 71:         "¥001¥006ボタン"        /* 初期値データ,中央寄せ */
 72:     },
 73:     {
 74:         0,                      /* ハンドル等       */
 75:         {16,62,16+128,74},      /* 境界 1文字くらい余分に領
域をとっておく*/
 76:         DT_EDTTXT+DT_DISABL,    /* 編集可能テキスト+未帰還属
性 */
 77:         32,                     /* 初期値データサイズ */
 78:         "¥024文字列を入力して"  /* 初期値データ 最大20文字 */
 79:     }
 80: };
 81: 
 82: /*
 83:       ダイアログを開く位置(中央よりも少し上にしてある)
 84: */
 85: rect dlBounds={ 384-128,256-64-20,384+128,256+64-20 };
 86: 
 87: 
 88: /*
 89:        ダイアログ内のアイテムの値(初期値)
 90: */
 91: int      dI3Value=0;
 92: int      dI4Value=0;
 93: char     dI6Value[21]="¥020文字列を入力して";
 94: 
 95: /*
 96:       フィルタ関数
 97: */
 98: MyFilter(Dialog,ev)
 99: dialog *Dialog;
100: event  *ev;
101: {
102:     point_t okbtn;
103: 
104:     if( ev->eWhat == E_KEYDOWN ){
105:         switch((short)(ev->eWhom)){
106:         case 9:
107:             ev->eWhom &= 0xffff0000;
108:             ev->eWhom |= 32;
109:             break;
110:         case 13:
111:             okbtn.p.x=384+128-10;   /* 取り消しボタン */
112:             okbtn.p.y=256+64-20-10;
113:             ev->eWhere=okbtn;
114:             ev->eWhat =E_MSLDOWN;
115:             break;
116:         }
117:     }
118:     return 0;
119: }
120: 
121: 
122: /*********************************************
123:       ダイアログを使う
124: *********************************************/
125: doDialog()
126: {
127:     dialog   *dialogPtr;
128:     dlgIList **dIHdl;
129:     int      ditem;
130: 
131:     short    DI_T;
132:     int      DI_H;
133:     rect     DI_R;
134: 
135:     int      tmpV3;     /* アイテム3のテンポラリな値 */
136:     int      tmpV4;     /* アイテム4のテンポラリな値 */
137: 
138: 
139: /* (1) 領域を確保する */
140: 
141:     dIHdl=(dialog**)MMChHdlNew( sizeof(dItemList) );
142:     if( dIHdl == NULL ){
143:         DMError(0x101,"領域確保に失敗しました。");
144:         return ( FALSE );
145:     }
146:         ;
147: /* (2) アイテムリストをコピーする */
148: 
149:     memcpy(*dIHdl,&dItemList,sizeof(dItemList));
150: 
151: /* (3) ウインドウをオープンする */
152: 
153:     dialogPtr=DMOpen(NULL,&dlBounds,DWINTITLE,TRUE,DWINDEFID,
154:             (window *)-1,FALSE,TSGetID(),dIHdl);
155:     if( dialogPtr == NULL ){
156:         MMHdlDispose(dIHdl);
157:         DMError(0x101,"ウインドウがオープンできません。");
158:         return( FALSE );
159:     }
160: 
161: /* (3-1) アイテムの初期値を入れる */
162: 
163: 
164:     DIGet(dialogPtr,3,&DI_T,&DI_H,&DI_R);
165:     tmpV3=dI3Value;
166:     CMValueSet(DI_H,tmpV3);
167: 
168:     DIGet(dialogPtr,4,&DI_T,&DI_H,&DI_R);
169:     tmpV4=dI4Value;
170:     CMValueSet(DI_H,tmpV4);
171: 
172:     DIGet(dialogPtr,6,&DI_T,&DI_H,&DI_R);
173:     DITSet(DI_T,DI_H,dI6Value);
174: 
175:     DMDraw(dialogPtr);       /* 描き直し */
176: 
177: /* (4) ボタンが押されるのを待つ */
178: 
179:     while(1){
180:         ditem=DMControl((void*)MyFilter );
181: 
182: /* (5) 選択されたアイテムに応じた処理をする */
183: 
184:         if(ditem==1 || ditem==2) break; /* OKか取り消しか */
185: 
186:         DIGet(dialogPtr,ditem,&DI_T,&DI_H,&DI_R); /* ハンドル
を得る */
187:         if(ditem==3){
188:             tmpV3=CMValueGet(DI_H); /* 値を得る */
189:             tmpV3=(tmpV3==0)? 1 : 0;/* 値の変化 */
190:             CMValueSet(DI_H,tmpV3); /* 値を更新 */
191:             CMDrawOne(DI_H);    /* 描き直す */
192:         }
193:         else if(ditem==4){
194:             tmpV4=CMValueGet(DI_H);
195:             tmpV4=(tmpV4==0)? 1 : 0;
196:             CMValueSet(DI_H,tmpV4);
197:             CMDrawOne(DI_H);
198:         }
199:     }
200: 
201: /* (5-1) OKならアイテムの値を更新する */
202: 
203:     if(ditem==1){
204:         dI3Value=tmpV3;
205:         dI4Value=tmpV4;
206:         DIGet(dialogPtr,6,&DI_T,&DI_H,&DI_R);
207:         DITGet(DI_T,DI_H,dI6Value);
208:     }
209: 
210: /* (6) ウインドウをクローズする */
211: 
212:     DMDispose(dialogPtr);
213: 
214: /* (7) 領域を解放する */
215: 
216:     MMHdlDispose(dIHdl);
217: 
218: };
```

リスト5　バグ取り

```
10 /* FSX.Xへのパッチプログラム
20 /* FSX.X Ver.1.02 専用です
30 /* Ver.1.10ではfseekの引数を1fe91->2da0f
40 fp=fopen("fsx.x","rw")
50 fseek(fp,&H1FE91,0)
60 fputc(&H52,fp)
70 fclose(fp)
```

マシン語カクテル in Z80's Bar

第22回——最後の手段を——

3回連続という形でお届けした光君のデバッガも，ブレイクポイント，ファイル操作，そして今回の逆アセンブルときて，いよいよ完成です。本当にご苦労さまでした。プログラムリストの中に光君の苦心のあとが見られる?

シナリオ：金子俊一
特別監修：浦川博之

♪カラン，コロ〜ン

源光(以下光)：こんにちは。プログラムを持ってきましたよ。

ようこ(以下Yo)：いらっしゃいませ，Z80's Barへようこそ。おひとり様ですか？　お席にご案内します。

光：ここはデニーズですか。

Yo：おしぼりをどうぞ。ご注文がお決まりになりましたら呼んでください。

マスター(以下M)：こらこら。覚えたからってむやみに使うもんじゃありません。

Yo：は〜い。

M：ところで光君，もうツケはなくなってますよ。

光：知ってますよ。この前の分はちゃんと純ちゃんのツケにしたんですから。

M：それじゃあ，いったい？

光：あのプログラムは未完成のままでしたからね。どうしても完成させたくて。

M：さすが光君，いい心掛けですよ。

長老(以下老)：ふぉっふぉっふぉ。さては逆アセンブルできるようになったわけじゃな。

光：そのとおりです。

老：トレース機能もつけたんかのう。

光：いや，メモリが足りなくって。

老：何バイトくらい余っておるんじゃ？

光：4Kバイトに収めるつもりだったんですが，5バイトしか余ってないんですよ。

老：ほほう，それはきびしいのう。

光：いや，実はいろいろありまして，その5バイトも使ってしまう予定なんですよ。

老：なんと1バイトも余っておらんのか。

光：ええ。その昔のデバッガ，「ZAID」に比べて1つひとつの機能が強化されてるうえに，ファイルの入出力をつけましたからね。

老：それでファイルサイズが同じなら，まあ立派なもんじゃろ。

光：いやいや。

Yo：ふたりだけで話してないで少しは教えてよ。

老：おお，こわこわ。ようこちゃんを怒らせるとあとが怖いからのう。

内部の話はあのねのね

光：ええと，ようこさん。Z80の内部はどんな感じで動いているか知ってます？

Yo：知らない（きっぱり)。

光：とは思いましたけどね。

Yo：ずいぶんなことをいってくれるじゃない。最近なんだか冷たいのね。

光：そんなことはないんだけどなあ。

Yo：その逃げ腰なところがアヤしいのよ。

光：なんかイヤなことでもあったんですか？

Yo：なんにもないわよ。それよりちゃんと説明してよ。

光：え〜と，Z80のM1サイクルって知ってます？

Yo：そんなに私をいじめることが楽しいの？

M：ようこちゃんもからみますねぇ。

光：M1サイクルっていうのはオペコード・フェッチ・サイクルのことなんです。

Yo：でも，オペコードうんちゃらサイクルの略じゃないみたいだけど。

光：M1サイクルはマシンサイクル1(Machine Cycle 1）の略になるんですよ。

Yo：だったら，MC1っていえばいいじゃない。

光：それじゃあMCハマーと区別がつかないじゃないですか。もっとも，そっちのMCはマスター・オブ・セレモニー（Mastaer of Ceremonies）の略だけど。

Yo：それで，マシンサイクルっていうのはなに？

光：Z80の動きを知りたかったら絶対に必要なものなんですよ。

1)　オペコード・フェッチ・サイクル
2)　メモリ・リード・サイクル
3)　メモリ・ライト・サイクル
4)　I/Oリード・サイクル
5)　I/Oライト・サイクル

の5つがあって，5が終わったら1に戻るようになってるんだ。これがマシンサイクルなんだけど，かならずしも5つあるとはかぎらないんだ。

Yo：どうして？

光：たとえば「PUSH HL」だったら，1のオペコード・フェッチ・サイクルの次に3のメモリ・ライト・サイクルがあって，また1に戻るんだ。

老：「POP HL」じゃったら1のあとに2がきて，また1に戻るわけじゃ。

光：そう，つまり関係ないマシンサイクルは実行しないようになっているわけだ。

Yo：それで？

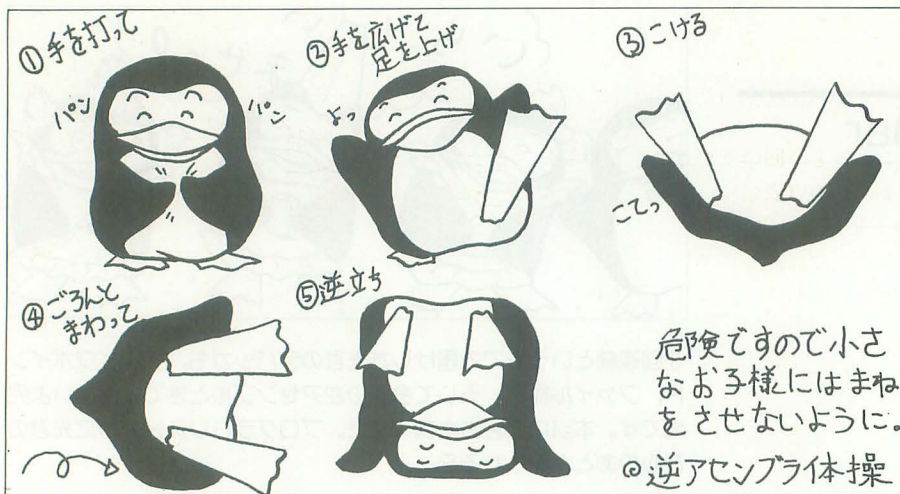
光：ところがオペコード・フェッチ，つまり命令を読み込むところはマシンサイクルの最初にかならずある。そこで，命令を読み込むサイクルをM1サイクルと呼ぶことになってるんだ。

Yo：ふうん。

光：このM1サイクルがわかれば，逆アセンブルができるようになるんだ。

Yo：結局それがいいたかったのね。

コマンドの説明　その3

○L　XXXX YYYY
　XXXX番地からYYYY番地までを逆アセンブルします。YYYY番地は省略できますが，省略するとエンドレスに逆アセンブルします。スペースキーで一時停止，ブレイクキーで停止します。プリンタスイッチに対応しています。
ex)　L 5000 5FFF

光：そう。これはとっても体系的に整理されているZ80の命令を見ればわかるよ。

老：ふぉっふぉっふぉ。どちらかといえば，Z80というよりも8080の命令が体系的なんじゃよ。

光：そうですね。でも＄CBで始まるビット操作命令あたりはかなりきれいですよ。

老：うむ。ただ，SLL命令がありそうでないのがイマイチじゃのう。

Yo：なんだかよくわからないわ。

光：それじゃあ代表的な命令をいくつか表1にまとめておきましょう。

老：ビット操作で命令判断ができるのがわかるじゃろうて。

スパゲッティおまち！

老：ところで，光君。そろそろプログラムを見せてもらいたいのじゃが。

光：うっ。

老：どうしたのじゃ。今日はプログラムを持ってきたんじゃろ？

光：そっそれがですねえ。

Yo：はい。スパゲッティのお客様。

光：ごめんなさい。

Yo：なにいってるのよ。お腹すいただろうからってマスターが作ってくれたのよ。

光：いや，僕の作ったプログラムもスパゲッティなんです。

老：おお，なんと。

光：いや，IX,IY命令さえなければそんなことはなかったんです。あるいはもう少しメモリに余裕があれば……。

老：さては，自己書き換えに手を出しおったな。

光：はっはっは。

Yo：開き直ってどうするのよ。

光：いや，もうしわけない。だけど，IX,IYが関わってこないところは十分にきれいなプログラムですよ。

老：それでは今回のプログラムのエッセンシャルコンディショニングでも聞かせてもらおうかのう。

M：長老，それをいうならエッセンス。ボケが多いですね，このごろ。

老：ほっときなさい。

光：えっと，今回のプログラムではジャンプテーブルというものを何回か使っています。

老：ほほう。

Yo：このリストではJP（HL）というのがいくつかあるみたいね。

光：そう，そこいらへんなんですよ。

Yo：この命令を解説してみましょう。

一同：おお～っ！

Yo：えっへん。これはHLレジスタが示すアドレスに入っているアドレスにジャンプするという……。

光＆老：ブブ～。

Yo：なにか違うの？

老：惜しいのう。

光：たしかに命令の書式を見るとそんな感じがするんだけど，実はHLレジスタが示すアドレスにジャンプするだけなんだ。

Yo：ずるい！

光：別に僕が決めたわけじゃないからね。

M：ところで光君，このリストをみるとサブルーチンからはリターンで戻ってきてるようですが。

光：いいところに目をつけましたね，マスター。

Yo：わかった。JP（HL）はスタックポインタに帰ってくるアドレスをプッシュしてから……。

光：まだまだ残念。スタックポインタの操作はプログラムでやっているんだよ。

Yo：ふ～ん。

光：さらに，スパゲッティとなっているIX,IYのあたりでは，戻り番地の操作をやってたりもする。

老：それはスパゲッチーじゃのう。

M：その発音はなんとかなりませんかね。

老：通じればいいんじゃ。

光：ともかく，逆アセンブルはちゃんとできますよ。

老：ほほう，かなりの自信じゃな。

光：なんたって，Z80のすべての命令を逆アセンブルして確かめましたからね。

Yo：ない命令を逆アセンブルするとどうなるの？

光：基本的には“？”マークを出すようにしたんですけどね。IX,IY系では無視される場合がありますね。

老：具体的にはどういうことかな？

光：知っている人も多いと思うんですけど，HL系の命令の前に＄DDを置くとIX系に，＄FDを置くとIY系になるんですよ。たとえば，＄21というのはLD HL,$XXXXという命令になるんです。

```
   21 34 12  : LD HL,$1234
DD 21 34 12  : LD IX,$1234
FD 21 34 12  : LD IY,$1234
```

表1　Z80の命令はこうなっている(代表例)

＊INC　r

0	0	← r →	1	0	0

r	rの値	ニーモニック	コード
B	000	INC B	$04
C	001	INC C	$0C
D	010	INC D	$14
E	011	INC E	$1C
H	100	INC H	$24
L	101	INC L	$2C
(HL)	110	INC (HL)	$34
A	111	INC A	$3C

＊SLA r

1	1	0	0	1	0	1	1

0	0	1	0	0	← r →

r	rの値	ニーモニック	コード
B	000	SLA B	$20
C	001	SLA C	$21
D	010	SLA D	$22
E	011	SLA E	$23
H	100	SLA H	$24
L	101	SLA L	$25
(HL)	110	SLA (HL)	$26
A	111	SLA A	$27

＊LD r, r'

0	1	← r →	← r' →

			r'							
			000	001	010	011	100	101	110	111
			B	C	D	E	H	L	(HL)	A
r	000	B	$40	$41	$42	$43	$44	$45	$46	$47
	001	C	$48	$49	$4A	$4B	$4C	$4D	$4E	$4F
	010	D	$50	$51	$52	$53	$54	$55	$56	$57
	011	E	$58	$59	$5A	$5B	$5C	$5D	$5E	$5F
	100	H	$60	$61	$62	$63	$64	$65	$66	$67
	101	L	$68	$69	$6A	$6B	$6C	$6D	$6E	$6F
	110	(HL)	$70	$71	$72	$73	$74	$75	$76	$77
	111	A	$78	$79	$7A	$7B	$7C	$7D	$7E	$7F

みたいな感じですね。

Yo：ふうん。

光：そこで＄DDが第１OPコードだったら，ふだんは「HL」と表示するサブルーチンを「IX」に変えてしまって，もう１度命令をフェッチさせるわけです。

老：そうなると21 34 12が待っているので，コンピュータはLD HL,$1234と表示しようとする。ところが「HL」は「IX」に変わってしまっているので，LD IX,$1234と表示するわけか。

光：はい。そのあとにまた「IX」を「HL」に戻します。

老：なるほど。

光：そこで，HLに関係ない命令が＄DDの後ろにあるとまるで意味がないんです。たとえば，＄01はLD BC,$＊＊＊＊なんですけど，

```
01 34 12      : LD BC,$1234
DD 01 34 12   : LD BC,$1234
FD 01 34 12   : LD BC,$1234
```

となってしまうんです。

老：なるほど。まあDD 01などという命令はZ80にはないから大丈夫じゃろう。

Yo：データの中に混じっていたときは？

光：暴走はしないはずだからOKです。

Yo：それじゃあ実際に動かしましょう。

ちゃんと動くね

光：えっと今回の変更点もまとめておきましょうかね（表2）。

Yo：なにか注意することはあるの？

光：そうだな。逆アセンブルの終了アドレスを設定しないと，64Kバイトのエンドレスループになるってことかな。

Yo：それって暴走っていうんじゃない？

光：違います。スペースキーで一時停止になるし，ブレイクキーで元に戻りますから。

Yo：なるほど。

光：あとはプリンタスイッチにも対応してますから，プリントアウトもできます。

老：ほう。ちゃんと動いておるではないか。行きあたりばったりでプログラム作ったわりには，ぴったり４Kバイトに収まっておるし。

光：ほっといてください。これでもZAIDに比べて３～８％程度のスピードアップになってるんですよ。

老：ところで，メモリサーチは途中で止められないようじゃが。

光：ギクッ！

表2　今月の変更点

5037	30	→	31
5039	38	→	30
503A	31	→	30
50CA	FF	→	74
50CB	53	→	56

老：まさか忘れておったわけではあるまいな？

光：そのまさかです。ソースリストで打ち込んだ人は先月号のリストの118行のFIND2というラベルと次の行の間にこの２行を挿入してください。

```
CALL    #PAUSE
DW      FIND5
```

これで一時停止や停止ができます。

Yo：この５バイトがあるからメモリが余ってないってわけなのね。

光：かたじけない。

M：いやあ，お疲れさまでした。今日はたあんと食べてってね。

光：それじゃ，ティラミスと３角フラスコ型のポストウォーターをお願いします。

Yo：ご注文を繰り返します。ケイジュンジャンバラヤがおひとつ……。

―つづく―

リスト1

```
0000                1 ;     Break Pointer part 3
0000                2 ;
0000                3 ;         by Hikaru Minamoto
0000                4
0000                5           OFFSET  $C674-$5674
5674                6           ORG     $5674
5674                7
5674                8 ;Label            Address   Break
5674                9
5674               10 #DSK     EQU      $1F5D    ; system work
5674               11 #STKAD   EQU      $1F6C    ;        :
5674               12 #EXADR   EQU      $1F6E    ;        :
5674               13 #DTADR   EQU      $1F70    ;        :
5674               14 #SIZE    EQU      $1F72    ;        :
5674               15 #KBFAD   EQU      $1F76    ;        :
5674               16 #PRCNT   EQU      $1F7A    ;        :
5674               17 #LPSW    EQU      $1F7C    ;        :
5674               18 #MON     EQU      $1F8E    ; nothing
5674               19 #FILE    EQU      $1FA3    ; AF,BC,DE,HL
5674               20 #RDD     EQU      $1FA6    ; AF,BC,DE,HL
5674               21 #WROD    EQU      $1FAC    ; AF,BC,DE,HL
5674               22 #WOPEN   EQU      $1FAF    ; AF,BC,DE,HL
5674               23 #HLHEX   EQU      $1FB2    ; AF,DE+4,HL
5674               24 #2HEX    EQU      $1FB5    ; AF,DE+2
5674               25 #HEX     EQU      $1FB8    ; AF
5674               26 #PRTHL   EQU      $1FBE    ; AF
5674               27 #PRTHX   EQU      $1FC1    ; AF
5674               28 #BELL    EQU      $1FC4    ; AF
5674               29 #PAUSE   EQU      $1FC7    ; AF
5674               30 #BRKEY   EQU      $1FCD    ; AF
5674               31 #GETL    EQU      $1FD3    ; AF
5674               32 #LPTOF   EQU      $1FD6    ; nothing
5674               33 #LPTON   EQU      $1FD9    ; nothing
5674               34 #TAB     EQU      $1FDF    ; AF
5674               35 #MPRINT  EQU      $1FE2    ; AF,DE
5674               36 #MSX     EQU      $1FE5    ; F
5674               37 #LTNL    EQU      $1FEE    ; nothing
5674               38 #PRINTS  EQU      $1FF1    ; F
5674               39 #PRINT   EQU      $1FF4    ; F
5674               40 #VER     EQU      $1FF7    ; HL
5674               41 #HOT     EQU      $1FFA    ; nothing
5674               42 #DIR     EQU      $2006    ; AF,BC,DE,HL
5674               43 #ROPEN   EQU      $2009    ; AF,BC,DE,HL
5674               44 #CSR     EQU      $2018    ; HL
5674               45 #SCRN    EQU      $201B    ; AF
5674               46 #FLGET   EQU      $2021    ; AF
5674               47 #RDVSW   EQU      $2024    ; A
5674               48 #SDVSW   EQU      $2027    ; AF
5674               49
5674               50 PRN      EQU      $516B
5674               51 DUMMY    EQU      $53FF
5674               52
5674               53 DISASM
5674 1A            54          LD       A,(DE)
5675 FE 20 20      55          IF A=" " THEN INC DE JR DISASM
5678 03 13 18
567B F8
567C CD B2 1F      56          CALL     #HLHEX
567F 22 E8 56      57          LD       (STADR),HL
5682 13            58          INC      DE
5683 CD B2 1F      59          CALL     #HLHEX
5686 30 03         60          JR       NC,DIS2
5688 21 FF FF      61          LD       HL,$FFFF  ; Endless loop
568B               62 DIS2
568B 22 EA 56      63          LD       (ENADR),HL
568E CD 6B 51      64          CALL     PRN
5691               65 DISJP
5691 CD EE 1F      66          CALL     #LTNL
5694 CD C7 1F      67          CALL     #PAUSE
5697 FF 53         68          DW       DUMMY   ;RET
5699 2A E8 56      69          LD       HL,(STADR)
569C ED 5B EA      70          LD       DE,(ENADR)
569F 56
56A0 37            71          SCF
56A1 ED 52         72          SBC      HL,DE
56A3 38 04         73          JR       C,DISJP2
56A5 CD D6 1F      74          CALL     #LPTOF
56A8 C9            75          RET
56A9               76 DISJP2
56A9 11 91 56      77          LD       DE,DISJP
56AC D5            78          PUSH     DE
56AD 2A E8 56      79          LD       HL,(STADR)
56B0 CD BE 1F      80          CALL     #PRTHL
56B3 CD F1 1F      81          CALL     #PRINTS
56B6               82 DISJP21
56B6 44 4D         83          LD       BC,HL
56B8 16 00         84          LD       D,0
56BA 5E            85          LD       E,(HL)
56BB 7B            86          LD       A,E
56BC FE 40         87          CP       $40
56BE 38 07         88          JR       C,DISJP3
56C0 FE C0         89          CP       $C0
56C2 38 09         90          JR       C,DISJP4
56C4 D6 80         91          SUB      $80
56C6 5F            92          LD       E,A
56C7               93 DISJP3
56C7 21 EC 56      94          LD       HL,JPTABLE
56CA 19            95          ADD      HL,DE
56CB 18 14         96          JR       TJP
56CD               97 DISJP4
56CD FE 76         98          CP       $76
56CF CA 85 5A      99          JP       Z,HALT
56D2 FE 80        100          CP       $80
56D4 DA B7 5C     101          JP       C,LDRR
56D7 1F 1F        102          RRA    :RRA
56D9 E6 0E        103          AND      14
56DB 21 EC 57     104          LD       HL,JPTABLE2
56DE              105 TABLEJP
56DE 16 00        106          LD       D,0
56E0 5F           107          LD       E,A
56E1              108 TJP
56E1 19           109          ADD      HL,DE
56E2 5E           110          LD       E,(HL)
56E3 23           111          INC      HL
```

```
56E4 56           112          LD     D,(HL)
56E5 62 6B        113          LD     HL,DE
56E7 E9           114          JP     (HL)
56E8              115 STADR
56E8 00 00        116          DS     2
56EA              117 ENADR
56EA 00 00        118          DS     2
56EC              119
56EC              120 JPTABLE
56EC              121
56EC              122 ; $0n
56EC AA 5A DF     123          DW  NOP :LDSS.NN :LD<SS>.A :INCSS
56EF 5C 65 5C
56F2 15 5C
56F4 06 5C B6     124          DW  INCR :DECR :LDR.N :RLCA
56F7 5B CC 5C
56FA EA 5A
56FC F1 5B 5D     125          DW  EXAF.AF' :ADD.SS :LDA.<SS> :DECSS
56FF 5B 95 5C
5702 C5 5B
5704 06 5C B6     126          DW  INCR :DECR :LDR.N :RRCA
5707 5B CC 5C
570A 0C 5B
570C              127 ; $1n
570C 5E 5A DF     128          DW  DJNZ :LDSS.NN :LD<SS>.A :INCSS
570F 5C 65 5C
5712 15 5C
5714 06 5C B6     129          DW  INCR :DECR :LDR.N :RLA
5717 5B CC 5C
571A DF 5A
571C 5C 5C 5D     130          DW  JRNN :ADD.SS :LDA.<SS> :DECSS
571F 5B 95 5C
5722 C5 5B
5724 06 5C B6     131          DW  INCR :DECR :LDR.N :RRA
5727 5B CC 5C
572A 01 5B
572C              132 ; $2n
572C 4A 5C DF     133          DW  JRCC :LDSS.NN :LD<NN>.HL :INCSS
572F 5C 87 5C
5732 15 5C
5734 06 5C B6     134          DW  INCR :DECR :LDR.N :DAA
5737 5B CC 5C
573A 49 5A
573C 4A 5C 5D     135          DW  JRCC :ADD.SS :LDHL.<NN> :DECSS
573F 5B F1 5C
5742 C5 5B
5744 06 5C B6     136          DW  INCR :DECR :LDR.N :CPLA
5747 5B CC 5C
574A 3E 5A
574C              137 ; $3n
574C 4A 5C DF     138          DW  JRCC :LDSS.NN :LD<NN>.A :INCSS
574F 5C 79 5C
5752 15 5C
5754 06 5C B6     139          DW  INCR :DECR :LDR.N :SCF
5757 5B CC 5C
575A 23 5B
575C 4A 5C 5D     140          DW  JRCC :ADD.SS :LDA.<NN> :DECSS
575F 5B A9 5C
5762 C5 5B
5764 06 5C B6     141          DW  INCR :DECR :LDR.N :CCF
5767 5B CC 5C
576A 33 5A
576C              142 ; $4n
576C              143 ; $5n
576C              144 ; $6n
576C              145 ; $7n
576C              146 ; $8n
576C              147 ; $9n
576C              148 ; $An
576C              149 ; $Bn
576C              150 ; $Cn
576C 3D 5D 24     151          DW  RETCC :POPSS :JPCC :JPNN
576F 5D 38 5C
5772 24 5C
5774 86 5B 35     152          DW  CALLCC :PUSHSS :ADD.N :RSTN
5777 5D 55 5B
577A 4C 5D
577C 3D 5D C3     153          DW  RETCC :RETT :JPCC :CB
577F 5A 38 5C
5782 D1 5D
5784 86 5B 98     154          DW  CALLCC :CALL :ADC.N :RSTN
5787 5B 3F 5B
578A 4C 5D
578C              155 ; $Dn
578C 3D 5D 24     156          DW  RETCC :POPSS :JPCC :OUT<N>.A
578F 5D 38 5C
5792 B5 5A
5794 86 5B 35     157          DW  CALLCC :PUSHSS :SUBN :RSTN
5797 5D 7D 5D
579A 4C 5D
579C 3D 5D 7A     158          DW  RETCC :EXX :JPCC :INA.<N>
579F 5A 38 5C
57A2 91 5A
57A4 86 5B 75     159          DW  CALLCC :IX :SBCA.N :RSTN
57A7 5F 6F 5D
57AA 4C 5D
57AC              160 ; $En
57AC 3D 5D 24     161          DW  RETCC :POPSS :JPCC :EX<SP>.HL
57AF 5D 38 5C
57B2 D4 5B
57B4 86 5B 35     162          DW  CALLCC :PUSHSS :ANDN :RSTN
57B7 5D 7D 5B
57BA 4C 5D
57BC 3D 5D 2D     163          DW  RETCC :JP<HL> :JPCC :EXDE.HL
57BF 5C 38 5C
57C2 E4 5B
57C4 86 5B 1E     164          DW  CALLCC :ED :XORN :RSTN
57C7 5E 85 5D
57CA 4C 5D
57CC              165 ; $Fn
57CC 3D 5D 24     166          DW  RETCC :POPSS :JPCC :DI
57CF 5D 38 5C
57D2 54 5A
57D4 86 5B 35     167          DW  CALLCC :PUSHSS :ORN :RSTN
57D7 5D 1B 5D
57DA 4C 5D
57DC 3D 5D FF     168          DW  RETCC :LDSP.HL :JPCC :EI
57DF 5C 38 5C
57E2 70 5A
57E4 86 5B 79     169          DW  CALLCC :IY :CPN :RSTN
57E7 5F AD 5B
57EA 4C 5D
57EC              170
57EC              171 JPTABLE2
57EC 4D 5B 2E     172          DW  ADD.R  :ADC.R  :SUB.R  :SBC.R
57EF 5B 9E 5D
57F2 8D 5D
57F4 71 5B A6     173          DW  ANDR  :XORR  :ORR  :CPR
57F7 5D 0F 5D
57FA A1 5B
57FC              174
57FC              175 TAB
57FC C5           176          PUSH   BC
57FD 06 1C        177          LD     B,28
57FF CD DF 1F     178          CALL   #TAB
5802 C1           179          POP    BC
5803 C9           180          RET
5804              181 KAKIKAE
5804 C9           182          RET
5805              183 BYTE4
5805 C5           184          PUSH   BC
5806 06 04        185          LD     B,4
5808 18 12        186          JR     BYTE
580A              187 BYTE3
580A C5           188          PUSH   BC
580B 06 03        189          LD     B,3
580D 18 0D        190          JR     BYTE
580F              191 BYTE2
580F C5           192          PUSH   BC
5810 06 02        193          LD     B,2
5812 18 08        194          JR     BYTE
5814              195 BYTE11
5814 C5           196          PUSH   BC
5815 06 01        197          LD     B,1
5817 18 03        198          JR     BYTE
5819              199 BYTE1
5819 C5           200          PUSH   BC
581A 06 01        201          LD     B,1
581C              202
581C              203 BYTE
581C 2A E8 56     204          LD     HL,(STADR)
581F              205 BYTE0
581F 7E           206          LD     A,(HL)
5820 CD C1 1F     207          CALL   #PRTHX
5823 CD F1 1F     208          CALL   #PRINTS
5826 23           209          INC    HL
5827 10 F6        210          DJNZ   BYTE0
5829 06 14        211          LD     B,20
582B CD DF 1F     212          CALL   #TAB
582E 22 E8 56     213          LD     (STADR),HL
5831 C1           214          POP    BC
5832 C9           215          RET
5833              216 RRA4
5833 1F 1F 1F     217          RRA  :RRA  :RRA  :RRA
5836 1F
5837 C9           218          RET
5838              219          ;
5838              220 N
5838 03           221          INC    BC
5839 3E 24        222          LD     A,"$"
583B CD F4 1F     223          CALL   #PRINT
583E 0A           224          LD     A,(BC)
583F CD C1 1F     225          CALL   #PRTHX
5842 C9           226          RET
5843              227 NN
5843 3E 24        228          LD     A,"$"
5845 CD F4 1F     229          CALL   #PRINT
5848 60 69        230          LD     HL,BC
584A 23           231          INC    HL
584B 4E           232          LD     C,(HL)
584C 23           233          INC    HL
584D 46           234          LD     B,(HL)
584E 60 69        235          LD     HL,BC
5850 CD BE 1F     236          CALL   #PRTHL
5853 C9           237          RET
5854              238 RJ
5854 3E 24        239          LD     A,"$"
5856 CD F4 1F     240          CALL   #PRINT
5859 03           241          INC    BC
585A 0A           242          LD     A,(BC)
585B 03           243          INC    BC
585C 6F           244          LD     L,A
585D 07           245          RLCA
585E 3E 00        246          LD     A,0 ; NOT(XOR A)
5860 9F           247          SBC    A,A
5861 67           248          LD     H,A ; H=00orFF
5862 09           249          ADD    HL,BC
5863 CD BE 1F     250          CALL   #PRTHL
5866 C9           251          RET
5867              252 CC
5867 21 1F 59     253          LD     HL,CCW
586A 18 3C        254          JR     R3+3
586C              255 CC.
586C CD 67 58     256          CALL   CC
586F 18 5A        257          JR     R.+3
5871              258 <N>
5871 3E 28        259          LD     A,"("
5873 CD F4 1F     260          CALL   #PRINT
5876 CD 38 58     261          CALL   N
5879 3E 29        262          LD     A,")"
587B CD F4 1F     263          CALL   #PRINT
587E C9           264          RET
587F              265 <NN>
587F 3E 28        266          LD     A,"("
5881 CD F4 1F     267          CALL   #PRINT
5884 CD 43 58     268          CALL   NN
5887 3E 29        269          LD     A,")"
5889 CD F4 1F     270          CALL   #PRINT
588C C9           271          RET
588D              272 <SS>
588D 08           273          EX     AF,AF'
588E 3E 28        274          LD     A,"("
5890 CD F4 1F     275          CALL   #PRINT
5893 08           276          EX     AF,AF'
5894 CD 9D 58     277          CALL   SS
5897 3E 29        278          LD     A,")"
5899 CD F4 1F     279          CALL   #PRINT
589C C9           280          RET
```

▶最近PIXY99Xを買いました（安くなったもので）。DSPの威力は絶大です。そういえばあの「PIXYの幅があれば人間は生きていける」という宣伝は私の部屋のことでは？
築城　光義(18)福島県

```
589D           281 SS
589D 21 00 59  282          LD      HL,REG2
58A0 18 06     283          JR      R3+3
58A2           284 R
58A2 00 00 00  285          DS      3       ; (IX+d)
58A5           286 R3
58A5 21 D1 58  287          LD      HL,REG1
58A8 16 00     288          LD      D,0
58AA 5F        289          LD      E,A
58AB 19        290          ADD     HL,DE
58AC 19        291          ADD     HL,DE
58AD 5E        292          LD      E,(HL)
58AE 23        293          INC     HL
58AF 56        294          LD      D,(HL)
58B0 CD E5 1F  295          CALL    #MSX
58B3 C9        296          RET
58B4           297 <N>.
58B4 CD 71 58  298          CALL    <N>
58B7 18 12     299          JR      R.+3
58B9           300 <NN>.
58B9 CD 7F 58  301          CALL    <NN>
58BC 18 0D     302          JR      R.+3
58BE           303 <SS>.
58BE CD 8D 58  304          CALL    <SS>
58C1 18 08     305          JR      R.+3
58C3           306 SS.
58C3 CD 9D 58  307          CALL    SS
58C6 18 03     308          JR      R.+3
58C8           309 R.
58C8 CD A2 58  310          CALL    R
58CB 3E 2C     311          LD      A,","
58CD CD F4 1F  312          CALL    #PRINT
58D0 C9        313          RET
58D1           314
58D1 E9 58 33  315 REG1     DW      B,C,D,E,H,L,<HL>,A,I,RR,<C>
58D4 59 EB 58
58D7 ED 58 EF
58DA 58 F1 58
58DD F3 58 E7
58E0 58 F8 58
58E3 FA 58 FC
58E6 58
58E7 41 00     316 A        DB      "A",0
58E9 42 00     317 B        DB      "B",0 ; C ∧ Carry
58EB 44 00     318 D        DB      "D",0
58ED 45 00     319 E        DB      "E",0
58EF 48 00     320 H        DB      "H",0
58F1 4C 00     321 L        DB      "L",0
58F3 28 48 4C  322 <HL>     DB      "(","H","L",")",0
58F6 29 00
58F8 49 00     323 I        DB    : "I",0
58FA 52 00     324 RR       DB      "R",0
58FC 28 43 29  325 <C>      DB      "(","C",")",0
58FF 00
5900           326          ;
5900 10 59 13  327 REG2     DW      BC,DE,HLL,SP,BC,DE,HLL,AF
5903 59 16 59
5906 19 59 10
5909 59 13 59
590C 16 59 1C
590F 59
5910 42 43 00  328 BC       DB      "B","C",0
5913 44 45 00  329 DE       DB      "D","E",0
5916 48 4C 00  330 HLL      DB      "H","L",0
5919 53 50 00  331 SP       DB      "S","P",0
591C 41 46 00  332 AF       DB      "A","F",0
591F           333          ;
591F 2F 59 30  334 CCW      DW      NZ,Z,NC,C,PO,PE,P,M
5922 59 32 59
5925 33 59 35
5928 59 38 59
592B 3B 59 3D
592E 59
592F 4E        335 NZ       DB      "N"
5930 5A 00     336 Z        DB      "Z",0
5932 4E        337 NC       DB      "N"
5933 43 00     338 C        DB      "C",0
5935 50 4F 00  339 PO       DB      "P","O",0
5938 50 45 00  340 PE       DB      "P","E",0
593B 50 00     341 P        DB      "P",0
593D 4D 00     342 M        DB      "M",0
593F           343          ;
593F           344 ADC
593F CD E2 1F  345          CALL    #MPRINT
5942 41 44 43  346          DB      "A","D","C",0
5945 00
5946 C3 FC 57  347          JP      TAB
5949           348 ADD
5949 CD E2 1F  349          CALL    #MPRINT
594C 41 44 44  350          DB      "A","D","D",0
594F 00
5950 C3 FC 57  351          JP      TAB
5953           352 AND
5953 CD E2 1F  353          CALL    #MPRINT
5956 41 4E 44  354          DB      "A","N","D",0
5959 00
595A C3 FC 57  355          JP      TAB
595D           356 CAL
595D CD E2 1F  357          CALL    #MPRINT
5960 43 41 4C  358          DB      "C","A","L","L",0
5963 4C 00
5965 C3 FC 57  359          JP      TAB
5968           360 CP
5968 CD E2 1F  361          CALL    #MPRINT
596B 43 50 00  362          DB      "C","P",0
596E C3 FC 57  363          JP      TAB
5971           364 DEC
5971 CD E2 1F  365          CALL    #MPRINT
5974 44 45 43  366          DB      "D","E","C",0
5977 00
5978 C3 FC 57  367          JP      TAB
597B           368 EX
597B CD E2 1F  369          CALL    #MPRINT
597E 45 58 00  370          DB      "E","X",0
5981 C3 FC 57  371          JP      TAB
5984           372 IM
5984 CD E2 1F  373          CALL    #MPRINT
5987 49 4D 00  374          DB      "I","M",0
598A C3 FC 57  375          JP      TAB
598D           376 IN
598D CD E2 1F  377          CALL    #MPRINT
5990 49 4E 00  378          DB      "I","N",0
5993 C3 FC 57  379          JP      TAB
5996           380 INC
5996 CD E2 1F  381          CALL    #MPRINT
5999 49 4E 43  382          DB      "I","N","C",0
599C 00
599D C3 FC 57  383          JP      TAB
59A0           384 JP
59A0 CD E2 1F  385          CALL    #MPRINT
59A3 4A 50 00  386          DB      "J","P",0
59A6 C3 FC 57  387          JP      TAB
59A9           388 JR
59A9 CD E2 1F  389          CALL    #MPRINT
59AC 4A 52 00  390          DB      "J","R",0
59AF C3 FC 57  391          JP      TAB
59B2           392 LD
59B2 CD E2 1F  393          CALL    #MPRINT
59B5 4C 44 00  394          DB      "L","D",0
59B8 C3 FC 57  395          JP      TAB
59BB           396 OR
59BB CD E2 1F  397          CALL    #MPRINT
59BE 4F 52 00  398          DB      "O","R",0
59C1 C3 FC 57  399          JP      TAB
59C4           400 OUT
59C4 CD E2 1F  401          CALL    #MPRINT
59C7 4F 55 54  402          DB      "O","U","T",0
59CA 00
59CB C3 FC 57  403          JP      TAB
59CE           404 POP
59CE CD E2 1F  405          CALL    #MPRINT
59D1 50 4F 50  406          DB      "P","O","P",0
59D4 00
59D5 C3 FC 57  407          JP      TAB
59D8           408 PUSH
59D8 CD E2 1F  409          CALL    #MPRINT
59DB 50 55 53  410          DB      "P","U","S","H",0
59DE 48 00
59E0 C3 FC 57  411          JP      TAB
59E3           412 RET
59E3 CD E2 1F  413          CALL    #MPRINT
59E6 52 45 54  414          DB      "R","E","T",0
59E9 00
59EA C3 FC 57  415          JP      TAB
59ED           416 RST
59ED CD E2 1F  417          CALL    #MPRINT
59F0 52 53 54  418          DB      "R","S","T",0
59F3 00
59F4 C3 FC 57  419          JP      TAB
59F7           420 SBC
59F7 CD E2 1F  421          CALL    #MPRINT
59FA 53 42 43  422          DB      "S","B","C",0
59FD 00
59FE C3 FC 57  423          JP      TAB
5A01           424 SUB
5A01 CD E2 1F  425          CALL    #MPRINT
5A04 53 55 42  426          DB      "S","U","B",0
5A07 00
5A08 C3 FC 57  427          JP      TAB
5A0B           428 XOR
5A0B CD E2 1F  429          CALL    #MPRINT
5A0E 58 4F 52  430          DB      "X","O","R",0
5A11 00
5A12 C3 FC 57  431          JP    : TAB
5A15           432 BIT
5A15 CD E2 1F  433          CALL    #MPRINT
5A18 42 49 54  434          DB      "B","I","T",0
5A1B 00
5A1C C3 FC 57  435          JP      TAB
5A1F           436 RES
5A1F CD E2 1F  437          CALL    #MPRINT
5A22 52 45 53  438          DB      "R","E","S",0
5A25 00
5A26 C3 FC 57  439          JP      TAB
5A29           440 SET
5A29 CD E2 1F  441          CALL    #MPRINT
5A2C 53 45 54  442          DB      "S","E","T",0
5A2F 00
5A30 C3 FC 57  443          JP      TAB
5A33           444          ;
5A33           445 CCF
5A33 CD 19 58  446          CALL    BYTE1
5A36 CD E2 1F  447          CALL    #MPRINT
5A39 43 43 46  448          DB      "C","C","F",0
5A3C 00
5A3D C9        449          RET
5A3E           450 CPLA
5A3E CD 19 58  451          CALL    BYTE1
5A41 CD E2 1F  452          CALL    #MPRINT
5A44 43 50 4C  453          DB      "C","P","L",0
5A47 00
5A48 C9        454          RET
5A49           455 DAA
5A49 CD 19 58  456          CALL    BYTE1
5A4C CD E2 1F  457          CALL    #MPRINT
5A4F 44 41 41  458          DB      "D","A","A",0
5A52 00
5A53 C9        459          RET
5A54           460 DI
5A54 CD 19 58  461          CALL    BYTE1
5A57 CD E2 1F  462          CALL    #MPRINT
5A5A 44 49 00  463          DB      "D","I",0
5A5D C9        464          RET
5A5E           465 DJNZ
5A5E CD 0F 58  466          CALL    BYTE2
5A61 CD E2 1F  467          CALL    #MPRINT
5A64 44 4A 4E  468          DB      "D","J","N","Z",0
5A67 5A 00
5A69 CD FC 57  469          CALL    TAB
5A6C CD 54 58  470          CALL    RJ
5A6F C9        471          RET
5A70           472 EI
5A70 CD 19 58  473          CALL    BYTE1
5A73 CD E2 1F  474          CALL    #MPRINT
5A76 45 49 00  475          DB      "E","I",0
5A79 C9        476          RET
5A7A           477 EXX
5A7A CD 19 58  478          CALL    BYTE1
```

▶最近，あまりMZをいじっていない……。久びさにアセンブラでもやろうかと思ったが無意識に，move.l a0,a1と書いてしまった。68000に慣れてしまうとZ80はなんて不自由なんだろう。思えば贅沢になったもんだ。　伊藤　直広(20)福島県

```
5A7D CD E2 1F   479          CALL    #MPRINT
5A80 45 58 58   480          DB      "E","X","X",0
5A83 00
5A84 C9         481          RET
5A85            482 HALT
5A85 CD 19 58   483          CALL    BYTE1
5A88 CD E2 1F   484          CALL    #MPRINT
5A8B 48 41 4C   485          DB      "H","A","L","T",0
5A8E 54 00
5A90 C9         486          RET
5A91            487 INA.<N>
5A91 CD 0F 58   488          CALL    BYTE2
5A94 CD 8D 59   489          CALL    IN
5A97 3E 07      490          LD      A,7
5A99 CD C8 58   491          CALL    R.
5A9C C3 71 58   492          JP      <N>
5A9F            493 NEG
5A9F CD 0F 58   494          CALL    BYTE2
5AA2 CD E2 1F   495          CALL    #MPRINT
5AA5 4E 45 47   496          DB      "N","E","G",0
5AA8 00
5AA9 C9         497          RET
5AAA            498 NOP
5AAA CD 19 58   499          CALL    BYTE1
5AAD CD E2 1F   500          CALL    #MPRINT
5AB0 4E 4F 50   501          DB      "N","O","P",0
5AB3 00
5AB4 C9         502          RET
5AB5            503 OUT<N>.A
5AB5 CD 0F 58   504          CALL    BYTE2
5AB8 CD C4 59   505          CALL    OUT
5ABB CD B4 58   506          CALL    <N>.
5ABE 3E 41      507          LD      A,'A'
5AC0 C3 F4 1F   508          JP      #PRINT
5AC3            509 RETT
5AC3 CD 19 58   510          CALL    BYTE1
5AC6 C3 E3 59   511          JP      RET
5AC9            512 RETN
5AC9 CD 0F 58   513          CALL    BYTE2
5ACC CD E3 59   514          CALL    RET
5ACF 3E 4E      515          LD      A,"N"
5AD1 C3 F4 1F   516          JP      #PRINT
5AD4            517 RETI
5AD4 CD 0F 58   518          CALL    BYTE2
5AD7 CD E3 59   519          CALL    RET
5ADA 3E 49      520          LD      A,"I"
5ADC C3 F4 1F   521          JP      #PRINT
5ADF            522 RLA
5ADF CD 19 58   523          CALL    BYTE1
5AE2 CD E2 1F   524          CALL    #MPRINT
5AE5 52 4C 41   525          DB      "R","L","A",0
5AE8 00
5AE9 C9         526          RET
5AEA            527 RLCA
5AEA CD 19 58   528          CALL    BYTE1
5AED CD E2 1F   529          CALL    #MPRINT
5AF0 52 4C 43   530          DB      "R","L","C","A",0
5AF3 41 00
5AF5 C9         531          RET
5AF6            532 RLD
5AF6 CD 0F 58   533          CALL    BYTE2
5AF9 CD E2 1F   534          CALL    #MPRINT
5AFC 52 4C 44   535          DB      "R","L","D",0
5AFF 00
5B00 C9         536          RET
5B01            537 RRA
5B01 CD 19 58   538          CALL    BYTE1
5B04 CD E2 1F   539          CALL    #MPRINT
5B07 52 52 41   540          DB      "R","R","A",0
5B0A 00
5B0B C9         541          RET
5B0C            542 RRCA
5B0C CD 19 58   543          CALL    BYTE1
5B0F CD E2 1F   544          CALL    #MPRINT
5B12 52 52 43   545          DB      "R","R","C","A",0
5B15 41 00
5B17 C9         546          RET
5B18            547 RRD
5B18 CD 0F 58   548          CALL    BYTE2
5B1B CD E2 1F   549          CALL    #MPRINT
5B1E 52 52 44   550          DB      "R","R","D",0
5B21 00
5B22 C9         551          RET
5B23            552 SCF
5B23 CD 19 58   553          CALL    BYTE1
5B26 CD E2 1F   554          CALL    #MPRINT
5B29 53 43 46   555          DB      "S","C","F",0
5B2C 00
5B2D C9         556          RET
5B2E            557
5B2E            558          ;
5B2E            559 ADC.R
5B2E CD 14 58   560          CALL    BYTE11
5B31 CD 3F 59   561          CALL    ADC
5B34 3E 07      562          LD      A,7
5B36 CD C8 58   563          CALL    R.
5B39 0A         564          LD      A,(BC)
5B3A E6 07      565          AND     7
5B3C C3 A2 58   566          JP      R
5B3F            567 ADC.N
5B3F CD 0F 58   568          CALL    BYTE2
5B42 CD 3F 59   569          CALL    ADC
5B45 3E 07      570          LD      A,7
5B47 CD C8 58   571          CALL    R.
5B4A C3 38 58   572          JP      N
5B4D            573 ADD.R
5B4D CD 14 58   574          CALL    BYTE11
5B50 CD 49 59   575          CALL    ADD
5B53 18 DF      576          JR      ADC.R+6
5B55            577 ADD.N
5B55 CD 0F 58   578          CALL    BYTE2
5B58 CD 49 59   579          CALL    ADD
5B5B 18 E8      580          JR      ADC.N+6
5B5D            581 ADD.SS
5B5D CL 19 58   582          CALL    BYTE1
5B60 CD 49 59   583          CALL    ADD
5B63 3E 02      584          LD      A,2
5B65 CD C3 58   585          CALL    SS.
```

```
5B68 0A         586          LD      A,(BC)
5B69 CD 33 58   587          CALL    RRA4
5B6C E6 03      588          AND     3
5B6E C3 9D 58   589          JP      SS
5B71            590 ANDR
5B71 CD 14 58   591          CALL    BYTE11
5B74 CD 53 59   592          CALL    AND
5B77 0A         593          LD      A,(BC)
5B78 E6 07      594          AND     7
5B7A C3 A2 58   595          JP      R
5B7D            596 ANDN
5B7D CD 0F 58   597          CALL    BYTE2
5B80 CD 53 59   598          CALL    AND
5B83 C3 38 58   599          JP      N
5B86            600 CALLCC
5B86 CD 0A 58   601          CALL    BYTE3
5B89 CD 5D 59   602          CALL    CAL
5B8C 0A         603          LD      A,(BC)
5B8D 1F 1F 1F   604          RRA  :RRA  :RRA
5B90 E6 07      605          AND     7
5B92 CD 6C 58   606          CALL    CC.
5B95 C3 43 58   607          JP      NN
5B98            608 CALL
5B98 CD 0A 58   609          CALL    BYTE3
5B9B CD 5D 59   610          CALL    CAL
5B9E C3 43 58   611          JP      NN
5BA1            612 CPR
5BA1 CD 14 58   613          CALL    BYTE11
5BA4 CD 68 59   614          CALL    CP
5BA7 0A         615          LD      A,(BC)
5BA8 E6 07      616          AND     7
5BAA C3 A2 58   617          JP      R
5BAD            618 CPN
5BAD CD 0F 58   619          CALL    BYTE2
5BB0 CD 68 59   620          CALL    CP
5BB3 C3 38 58   621          JP      N
5BB6            622 DECR
5BB6 CD 14 58   623          CALL    BYTE11
5BB9 CD 71 59   624          CALL    DEC
5BBC 0A         625          LD      A,(BC)
5BBD 1F 1F 1F   626          RRA  :RRA  :RRA
5BC0 E6 07      627          AND     7
5BC2 C3 A2 58   628          JP      R
5BC5            629 DECSS
5BC5 CD 19 58   630          CALL    BYTE1
5BC8 CD 71 59   631          CALL    DEC
5BCB 0A         632          LD      A,(BC)
5BCC CD 33 58   633          CALL    RRA4
5BCF E6 07      634          AND     7
5BD1 C3 9D 58   635          JP      SS
5BD4            636 EX<SP>.HL
5BD4 CD 19 58   637          CALL    BYTE1
5BD7 CD 7B 59   638          CALL    EX
5BDA 3E 03      639          LD      A,3
5BDC CD BE 58   640          CALL    <SS>.
5BDF            641 EXHL
5BDF 3E 02      642          LD      A,2
5BE1 C3 9D 58   643          JP      SS
5BE4            644 EXDE.HL
5BE4 CD 19 58   645          CALL    BYTE1
5BE7 CD 7B 59   646          CALL    EX
5BEA 3E 01      647          LD      A,1
5BEC CD C3 58   648          CALL    SS.
5BEF 18 EE      649          JR      EXHL
5BF1            650 EXAF.AF'
5BF1 CD 19 58   651          CALL    BYTE1
5BF4 CD 7B 59   652          CALL    EX
5BF7 3E 07      653          LD      A,7
5BF9 CD C3 58   654          CALL    SS.
5BFC 3E 07      655          LD      A,7
5BFE CD 9D 58   656          CALL    SS
5C01 3E 27      657          LD      A,"'"
5C03 C3 F4 1F   658          JP      #PRINT
5C06            659 INCR
5C06 CD 14 58   660          CALL    BYTE11
5C09 CD 96 59   661          CALL    INC
5C0C 0A         662          LD      A,(BC)
5C0D 1F 1F 1F   663          RRA  :RRA  :RRA
5C10 E6 07      664          AND     7
5C12 C3 A2 58   665          JP      R
5C15            666 INCSS
5C15 CD 19 58   667          CALL    BYTE1
5C18 CD 96 59   668          CALL    INC
5C1B 0A         669          LD      A,(BC)
5C1C CD 33 58   670          CALL    RRA4
5C1F E6 07      671          AND     7
5C21 C3 9D 58   672          JP      SS
5C24            673 JPNN
5C24 CD 0A 58   674          CALL    BYTE3
5C27 CD A0 59   675          CALL    JP
5C2A C3 43 58   676          JP      NN
5C2D            677 JP<HL>
5C2D CD 19 58   678          CALL    BYTE1
5C30 CD A0 59   679          CALL    JP
5C33 3E 02      680          LD      A,2
5C35 C3 8D 58   681          JP      <SS>
5C38            682 JPCC
5C38 CD 0A 58   683          CALL    BYTE3
5C3B CD A0 59   684          CALL    JP
5C3E 0A         685          LD      A,(BC)
5C3F 1F 1F 1F   686          RRA  :RRA  :RRA
5C42 E6 07      687          AND     7
5C44 CD 6C 58   688          CALL    CC.
5C47 C3 43 58   689          JP      NN
5C4A            690 JRCC
5C4A CD 0F 58   691          CALL    BYTE2
5C4D CD A9 59   692          CALL    JR
5C50 0A         693          LD      A,(BC)
5C51 1F 1F 1F   694          RRA  :RRA  :RRA
5C54 E6 03      695          AND     3
5C56 CD 6C 58   696          CALL    CC.
5C59 C3 54 58   697          JP      RJ
5C5C            698 JRNN
5C5C CD 0F 58   699          CALL    BYTE2
5C5F CD A9 59   700          CALL    JR
5C62 C3 54 58   701          JP      RJ
5C65            702 LD<SS>.A
```


▶「ちゃだワ」,「STUDIO X」,「イラスト」の３段攻撃の酒井強さんは，もう「ちゃだワ」の顔ですね。私なんか６回も参加しているのに１回も載ったことがないんですよ。シクシク……。
坊農　誠(20)福井県

```
5C65 CD 19 58   703            CALL    BYTE1
5C68 CD B2 59   704            CALL    LD
5C6B 0A         705            LD      A,(BC)
5C6C CD 33 58   706            CALL    RRA4
5C6F E6 01      707            AND     1
5C71 CD BE 58   708            CALL    <SS>.
5C74 3E 41      709            LD      A,"A"
5C76 C3 F4 1F   710            JP      #PRINT
5C79            711 LD<NN>.A
5C79 CD 0A 58   712            CALL    BYTE3
5C7C CD B2 59   713            CALL    LD
5C7F CD B9 58   714            CALL    <NN>.
5C82 3E 41      715            LD      A,"A"
5C84 C3 F4 1F   716            JP      #PRINT
5C87            717 LD<NN>.HL
5C87 CD 0A 58   718            CALL    BYTE3
5C8A CD B2 59   719            CALL    LD
5C8D CD B9 58   720            CALL    <NN>.
5C90 3E 02      721            LD      A,2
5C92 C3 9D 58   722            JP      SS
5C95            723 LDA.<SS>
5C95 CD 19 58   724            CALL    BYTE1
5C98 CD B2 59   725            CALL    LD
5C9B 3E 07      726            LD      A,7
5C9D CD C8 58   727            CALL    R.
5CA0 0A         728            LD      A,(BC)
5CA1 CD 33 58   729            CALL    RRA4
5CA4 E6 01      730            AND     1
5CA6 C3 8D 58   731            JP      <SS>
5CA9            732 LDA.<NN>
5CA9 CD 0A 58   733            CALL    BYTE3
5CAC CD B2 59   734            CALL    LD
5CAF 3E 07      735            LD      A,7
5CB1 CD C8 58   736            CALL    R.
5CB4 C3 7F 58   737            JP      <NN>
5CB7            738 LDRR
5CB7 CD 14 58   739            CALL    BYTE11
5CBA CD B2 59   740            CALL    LD
5CBD 0A         741            LD      A,(BC)
5CBE 1F 1F 1F   742            RRA   :RRA  :RRA
5CC1 E6 07      743            AND     7
5CC3 CD C8 58   744            CALL    R.
5CC6 0A         745            LD      A,(BC)
5CC7 E6 07      746            AND     7
5CC9 C3 A2 58   747            JP      R
5CCC            748 LDR.N
5CCC CD 0F 58   749            CALL    BYTE2
5CCF CD B2 59   750            CALL    LD
5CD2 0A         751            LD      A,(BC)
5CD3 1F 1F 1F   752            RRA   :RRA  :RRA
5CD6 E6 07      753            AND     7
5CD8 CD C8 58   754            CALL    R.
5CDB            755 LDR.NIX
5CDB 00         756            NOP
5CDC C3 38 58   757            JP      N
5CDF            758 LDSS.NN
5CDF CD 0A 58   759            CALL    BYTE3
5CE2 CD B2 59   760            CALL    LD
5CE5 0A         761            LD      A,(BC)
5CE6 CD 33 58   762            CALL    RRA4
5CE9 E6 03      763            AND     3
5CEB CD C3 58   764            CALL    SS.
5CEE C3 43 58   765            JP      NN
5CF1            766 LDHL.<NN>
5CF1 CD 0A 58   767            CALL    BYTE3
5CF4 CD B2 59   768            CALL    LD
5CF7 3E 02      769            LD      A,2
5CF9 CD C3 58   770            CALL    SS.
5CFC C3 7F 58   771            JP      <NN>
5CFF            772 LDSP.HL
5CFF CD 19 58   773            CALL    BYTE1
5D02 CD B2 59   774            CALL    LD
5D05 3E 03      775            LD      A,3
5D07 CD C3 58   776            CALL    SS.
5D0A 3E 02      777            LD      A,2
5D0C C3 9D 58   778            JP      SS
5D0F            779 ORR
5D0F CD 14 58   780            CALL    BYTE11
5D12 CD BB 59   781            CALL    OR
5D15 0A         782            LD      A,(BC)
5D16 E6 07      783            AND     7
5D18 C3 A2 58   784            JP      R
5D1B            785 ORN
5D1B CD 0F 58   786            CALL    BYTE2
5D1E CD BB 59   787            CALL    OR
5D21 C3 38 58   788            JP      N
5D24            789 POPSS
5D24 CD 19 58   790            CALL    BYTE1
5D27 CD CE 59   791            CALL    POP
5D2A            792 PSS
5D2A 0A         793            LD      A,(BC)
5D2B CD 33 58   794            CALL    RRA4
5D2E E6 03      795            AND     3
5D30 F6 04      796            OR      4
5D32 C3 9D 58   797            JP      SS
5D35            798 PUSHSS
5D35 CD 19 58   799            CALL    BYTE1
5D38 CD D8 59   800            CALL    PUSH
5D3B 18 ED      801            JR      PSS
5D3D            802 RETCC
5D3D CD 19 58   803            CALL    BYTE1
5D40 CD E3 59   804            CALL    RET
5D43 0A         805            LD      A,(BC)
5D44 1F 1F 1F   806            RRA   :RRA  :RRA
5D47 E6 07      807            AND     7
5D49 C3 67 58   808            JP      CC
5D4C            809 RSTN
5D4C CD 19 58   810            CALL    BYTE1
5D4F CD ED 59   811            CALL    RST
5D52 3E 24      812            LD      A,"$"
5D54 CD F4 1F   813            CALL    #PRINT
5D57 0A         814            LD      A,(BC)
5D58 1F 1F 1F   815            RRA   :RRA  :RRA
5D5B E6 07      816            AND     7
5D5D 08         817            EX      AF,AF'
5D5E AF         818            XOR     A
5D5F 08         819            EX      AF,AF'
5D60 1E 08      820            LD      E,8
```

```
5D62            821 RST1
5D62 B7         822            OR      A
5D63 28 06      823            JR      Z,RST2
5D65 08         824            EX      AF,AF'
5D66 83         825            ADD     A,E
5D67 08         826            EX      AF,AF'
5D68 3D         827            DEC     A
5D69 18 F7      828            JR      RST1
5D6B            829 RST2
5D6B 08         830            EX      AF,AF'
5D6C C3 C1 1F   831            JP      #PRTHX
5D6F            832 SBCA.N
5D6F CD 0F 58   833            CALL    BYTE2
5D72 CD F7 59   834            CALL    SBC
5D75            835 SBN
5D75 3E 07      836            LD      A,7
5D77 CD C8 58   837            CALL    R.
5D7A C3 38 58   838            JP      N
5D7D            839 SUBN
5D7D CD 0F 58   840            CALL    BYTE2
5D80 CD 01 5A   841            CALL    SUB
5D83 18 F0      842            JR      SBN
5D85            843 XORN
5D85 CD 0F 58   844            CALL    BYTE2
5D88 CD 0B 5A   845            CALL    XOR
5D8B 18 E8      846            JR      SBN
5D8D            847 SBC.R
5D8D CD 14 58   848            CALL    BYTE11
5D90 CD F7 59   849            CALL    SBC
5D93 3E 07      850            LD      A,7
5D95 CD C8 58   851            CALL    R.
5D98            852 SBR
5D98 0A         853            LD      A,(BC)
5D99 E6 07      854            AND     7
5D9B C3 A2 58   855            JP      R
5D9E            856 SUB.R
5D9E CD 14 58   857            CALL    BYTE11
5DA1 CD 01 5A   858            CALL    SUB
5DA4 18 F2      859            JR      SBR
5DA6            860 XORR
5DA6 CD 14 58   861            CALL    BYTE11
5DA9 CD 0B 5A   862            CALL    XOR
5DAC 18 EA      863            JR      SBR
5DAE            864 BITB.R
5DAE CD 15 5A   865            CALL    BIT
5DB1 79         866            LD      A,C
5DB2 1F 1F 1F   867            RRA   :RRA  :RRA
5DB5 E6 07      868            AND     7
5DB7 F6 30      869            OR      $30
5DB9 CD F4 1F   870            CALL    #PRINT
5DBC 3E 2C      871            LD      A,","
5DBE CD F4 1F   872            CALL    #PRINT
5DC1 79         873            LD      A,C
5DC2 E6 07      874            AND     7
5DC4 C3 A2 58   875            JP      R
5DC7            876 RESB.R
5DC7 CD 1F 5A   877            CALL    RES
5DCA 18 E5      878            JR      BITB.R+3
5DCC            879 SETB.R
5DCC CD 29 5A   880            CALL    SET
5DCF 18 E0      881            JR      BITB.R+3
5DD1            882 CB
5DD1 CD 0F 58   883            CALL    BYTE2
5DD4 00         884            NOP
5DD5 03         885            INC     BC
5DD6 0A         886            LD      A,(BC)
5DD7 4F         887            LD      C,A
5DD8 07 07      888            RLCA  :RLCA
5DDA E6 03      889            AND     3
5DDC 3D         890            DEC     A
5DDD 28 CF      891            JR      Z,BITB.R
5DDF 3D         892            DEC     A
5DE0 28 E5      893            JR      Z,RESB.R
5DE2 3D         894            DEC     A
5DE3 28 E7      895            JR      Z,SETB.R
5DE5            896            ;
5DE5 79         897            LD      A,C
5DE6 1F         898            RRA
5DE7 E6 1C      899            AND     28
5DE9 21 FE 5D   900            LD      HL,BITW
5DEC 16 00      901            LD      D,0
5DEE 5F         902            LD      E,A
5DEF 19         903            ADD     HL,DE
5DF0 54 5D      904            LD      DE,HL
5DF2 CD E5 1F   905            CALL    #MSX
5DF5 CD FC 57   906            CALL    TAB
5DF8 79         907            LD      A,C
5DF9 E6 07      908            AND     7
5DFB C3 A2 58   909            JP      R
5DFE            910 BITW
5DFE 52 4C 43   911            DB      "R","L","C",0
5E01 00
5E02 52 52 43   912            DB      "R","R","C",0
5E05 00
5E06 52 4C 20   913            DB      "R","L"," ",0
5E09 00
5E0A 52 52 20   914            DB      "R","R"," ",0
5E0D 00
5E0E 53 4C 41   915            DB      "S","L","A",0
5E11 00
5E12 53 52 41   916            DB      "S","R","A",0
5E15 00
5E16 3F 3F 3F   917            DB      "?","?","?",0
5E19 00
5E1A 53 52 4C   918            DB      "S","R","L",0
5E1D 00
5E1E            919 ED
5E1E 03         920            INC     BC
5E1F 0A         921            LD      A,(BC)
5E20 4F         922            LD      C,A
5E21            923            ;
5E21 FE 44      924            CP      $44
5E23 CA 9F 5A   925            JP      Z,NEG
5E26 FE 45      926            CP      $45
5E28 CA C9 5A   927            JP      Z,RETN
5E2B FE 4D      928            CP      $4D
5E2D CA D4 5A   929            JP      Z,RETI
5E30 FE 67      930            CP      $67
```

▶仕事のためにPC-286を買った。どんどん稼がせてX68000XVIの資金にしなくっちゃ。
なんだか妻に働かせて愛人に貢いでいるような気分。ま，いいか。　渡辺　仁(36)京都府

```
5E32 CA 18 5B   931              JP     Z,RRD
5E35 FE 6F      932              CP     $6F
5E37 CA F6 5A   933              JP     Z,RLD
5E3A E6 CF      934              AND    $CF
5E3C FE 42      935              CP     $42
5E3E CA C5 5E   936              JP     Z,SBC.SS
5E41 FE 43      937              CP     $43
5E43 CA D9 5E   938              JP     Z,LD<NN>.SS
5E46 FE 4A      939              CP     $4A
5E48 28 73      940              JR     Z,ADC.SS
5E4A FE 4B      941              CP     $4B
5E4C CA F4 5E   942              JP     Z,LDSS.<NN>
5E4F E6 F7      943              AND    $F7
5E51 FE 40      944              CP     $40
5E53 CA 0D 5F   945              JP     Z,INR.<C>
5E56 FE 41      946              CP     $41
5E58 CA 21 5F   947              JP     Z,OUT<C>.R
5E5B 79         948              LD     A,C
5E5C E6 E7      949              AND    $E7
5E5E FE 46      950              CP     $46
5E60 28 46      951              JR     Z,IMN
5E62 FE 47      952              CP     $47
5E64 28 16      953              JR     Z,LDIRA
5E66 FE 57      954              CP     $57
5E68 28 28      955              JR     Z,LDAIR
5E6A E6 FC      956              AND    $FC
5E6C FE A0      957              CP     $A0
5E6E CA 35 5F   958              JP     Z,LCIO
5E71            959              ;
5E71 CD 19 58   960              CALL   BYTE1
5E74 CD E2 1F   961              CALL   #MPRINT
5E77 3F 3F 3F   962              DB     "?","?","?",0
5E7A 00
5E7B C9         963              RET
5E7C            964 LDIRA
5E7C CD 0F 58   965              CALL   BYTE2
5E7F CD B2 59   966              CALL   LD
5E82 79         967              LD     A,C
5E83 1F 1F 1F   968              RRA  :RRA  :RRA
5E86 E6 01      969              AND    1
5E88 F6 08      970              OR     8
5E8A CD C8 58   971              CALL   R.
5E8D 3E 41      972              LD     A,"A"
5E8F C3 F4 1F   973              JP     #PRINT
5E92            974 LDAIR
5E92 CD 0F 58   975              CALL   BYTE2
5E95 CD B2 59   976              CALL   LD
5E98 3E 07      977              LD     A,7
5E9A CD C8 58   978              CALL   R.
5E9D 79         979              LD     A,C
5E9E 1F 1F 1F   980              RRA  :RRA  :RRA
5EA1 E6 01      981              AND    1
5EA3 F6 08      982              OR     8
5EA5 C3 A2 58   983              JP     R
5EA8            984 IMN
5EA8 CD 0F 58   985              CALL   BYTE2
5EAB CD 84 59   986              CALL   IM
5EAE 79         987              LD     A,C
5EAF 1F 1F 1F   988              RRA  :RRA  :RRA
5EB2 E6 03      989              AND    3
5EB4 20 01      990              JR     NZ,IMN2
5EB6 3C         991              INC    A
5EB7            992 IMN2
5EB7 3D         993              DEC    A
5EB8 F6 30      994              OR     $30
5EBA C3 F4 1F   995              JP     #PRINT
5EBD            996 ADC.SS
5EBD CD 0F 58   997              CALL   BYTE2
5EC0 CD 3F 59   998              CALL   ADC
5EC3 18 06      999              JR     SBC.SS+6
5EC5           1000 SBC.SS
5EC5 CD 0F 58  1001              CALL   BYTE2
5EC8 CD F7 59  1002              CALL   SBC
5ECB 3E 02     1003              LD     A,2
5ECD CD C3 58  1004              CALL   SS.
5ED0 79        1005              LD     A,C
5ED1 CD 33 58  1006              CALL   RRA4
5ED4 E6 03     1007              AND    3
5ED6 C3 9D 58  1008              JP     SS
5ED9           1009 LD<NN>.SS
5ED9 CD 05 58  1010              CALL   BYTE4
5EDC CD B2 59  1011              CALL   LD
5EDF C5        1012              PUSH   BC
5EE0 ED 4B E8  1013              LD     BC,(STADR)
5EE3 56
5EE4 0B        1014              DEC    BC
5EE5 0B        1015              DEC    BC
5EE6 0B        1016              DEC    BC
5EE7 CD B9 58  1017              CALL   <NN>.
5EEA C1        1018              POP    BC
5EEB 79        1019              LD     A,C
5EEC CD 33 58  1020              CALL   RRA4
5EEF E6 03     1021              AND    3
5EF1 C3 9D 58  1022              JP     SS
5EF4           1023 LDSS.<NN>
5EF4 CD 05 58  1024              CALL   BYTE4
5EF7 CD B2 59  1025              CALL   LD
5EFA 79        1026              LD     A,C
5EFB CD 33 58  1027              CALL   RRA4
5EFE E6 03     1028              AND    3
5F00 CD C3 58  1029              CALL   SS.
5F03 ED 4B E8  1030              LD     BC,(STADR)
5F06 56
5F07 0B        1031              DEC    BC
5F08 0B        1032              DEC    BC
5F09 0B        1033              DEC    BC
5F0A C3 7F 58  1034              JP     <NN>
5F0D           1035 INR.<C>
5F0D CD 0F 58  1036              CALL   BYTE2
5F10 CD 8D 59  1037              CALL   IN
5F13 79        1038              LD     A,C
5F14 1F 1F 1F  1039              RRA  :RRA  :RRA
5F17 E6 07     1040              AND    7
5F19 CD C8 58  1041              CALL   R.
5F1C 3E 0A     1042              LD     A,10
5F1E C3 A2 58  1043              JP     R
5F21           1044 OUT<C>.R
5F21 CD 0F 58  1045              CALL   BYTE2
5F24 CD C4 59  1046              CALL   OUT
5F27 3E 0A     1047              LD     A,10
5F29 CD C8 58  1048              CALL   R.
5F2C 79        1049              LD     A,C
5F2D 1F 1F 1F  1050              RRA  :RRA  :RRA
5F30 E6 07     1051              AND    7
5F32 C3 A2 58  1052              JP     R
5F35           1053 LCIO
5F35 3E 18     1054              LD     A,$18   ; LD A,JR
5F37 32 FC 57  1055              LD     (TAB),A ; !!!
5F3A           1056              ;
5F3A CD 0F 58  1057              CALL   BYTE2
5F3D 11 53 5F  1058              LD     DE,LCIO3
5F40 D5        1059              PUSH   DE
5F41 79        1060              LD     A,C
5F42 E6 03     1061              AND    3
5F44 17        1062              RLA
5F45 21 4B 5F  1063              LD     HL,LCIO2
5F48 C3 DE 56  1064              JP     TABLEJP
5F4B           1065 LCIO2
5F4B B2 59 68  1066              DW     LD  :CP  :IN  :OUT
5F4E 59 8D 59
5F51 C4 59
5F53           1067 LCIO3
5F53 79        1068              LD     A,C
5F54 CD 33 58  1069              CALL   RRA4
5F57 4F        1070              LD     C,A
5F58 38 07     1071              JR     C,DD
5F5A 3E 49     1072              LD     A,"I"
5F5C CD F4 1F  1073              CALL   #PRINT
5F5F 18 05     1074              JR     LCIO4
5F61           1075 DD
5F61 3E 44     1076              LD     A,"D"
5F63 CD F4 1F  1077              CALL   #PRINT
5F66           1078 LCIO4
5F66 79        1079              LD     A,C
5F67 1F        1080              RRA
5F68 30 05     1081              JR     NC,LCIO5
5F6A 3E 52     1082              LD     A,"R"
5F6C CD F4 1F  1083              CALL   #PRINT
5F6F           1084 LCIO5
5F6F 3E C5     1085              LD     A,$C5
5F71 32 FC 57  1086              LD     (TAB),A
5F74 C9        1087              RET
5F75           1088
5F75           1089 IX
5F75 3E 58     1090              LD     A,"X"
5F77 18 02     1091              JR     IXY
5F79           1092 IY
5F79 3E 59     1093              LD     A,"Y"
5F7B           1094 IXY
5F7B 32 17 59  1095              LD     (HLL+1),A
5F7E 3E 49     1096              LD     A,"I"
5F80 32 16 59  1097              LD     (HLL),A
5F83 3E C3     1098              LD     A,$C3
5F85 32 A2 58  1099              LD     (R),A
5F88 21 DE 5F  1100              LD     HL,<IXYD>
5F8B 22 A3 58  1101              LD     (R+1),HL
5F8E 21 16 58  1102              LD     HL,BYTE11+2
5F91 34        1103              INC    (HL)
5F92 21 11 58  1104              LD     HL,BYTE2+2
5F95 34        1105              INC    (HL)
5F96 3E 03     1106              LD     A,$03
5F98 32 D4 5D  1107              LD     (CB+3),A
5F9B 32 DB 5C  1108              LD     (LDR.NIX),A
5F9E           1109              ;
5F9E 2A E8 56  1110              LD     HL,(STADR)
5FA1 7E        1111              LD     A,(HL)
5FA2 CD C1 1F  1112              CALL   #PRTHX
5FA5 CD F1 1F  1113              CALL   #PRINTS
5FA8 23        1114              INC    HL
5FA9 22 E8 56  1115              LD     (STADR),HL
5FAC           1116              ;
5FAC 23        1117              INC    HL
5FAD 7E        1118              LD     A,(HL)
5FAE 32 FF 5F  1119              LD     (IXYD),A
5FB1 2B        1120              DEC    HL
5FB2           1121              ;
5FB2 D1        1122              POP    DE
5FB3 11 BA 5F  1123              LD     DE,IXY2
5FB6 D5        1124              PUSH   DE
5FB7 C3 B6 56  1125              JP     DISJP21
5FBA           1126 IXY2
5FBA 21 11 58  1127              LD     HL,BYTE2+2
5FBD 35        1128              DEC    (HL)
5FBE 21 16 58  1129              LD     HL,BYTE11+2
5FC1 35        1130              DEC    (HL)
5FC2 AF        1131              XOR    A
5FC3 32 DB 5C  1132              LD     (LDR.NIX),A
5FC6 32 D4 5D  1133              LD     (CB+3),A
5FC9 21 A2 58  1134              LD     HL,R
5FCC 77        1135              LD     (HL),A
5FCD 23        1136              INC    HL
5FCE 77        1137              LD     (HL),A
5FCF 23        1138              INC    HL
5FD0 77        1139              LD     (HL),A
5FD1 3E 48     1140              LD     A,"H"
5FD3 32 16 59  1141              LD     (HLL),A
5FD6 3E 4C     1142              LD     A,"L"
5FD8 32 17 59  1143              LD     (HLL+1),A
5FDB C3 91 56  1144              JP     DISJP
5FDE           1145 <IXYD>
5FDE FE 06     1146              CP     6
5FE0 C2 A5 58  1147              JP     NZ,R3
5FE3 3E 28     1148              LD     A,"("
5FE5 CD F4 1F  1149              CALL   #PRINT
5FE8 3E 02     1150              LD     A,2
5FEA CD 9D 58  1151              CALL   SS
5FED CD E2 1F  1152              CALL   #MPRINT
5FF0 2B 24 00  1153              DB     "+","$",0
5FF3 3A FF 5F  1154              LD     A,(IXYD)
5FF6 CD C1 1F  1155              CALL   #PRTHX
5FF9 3E 29     1156              LD     A,")"
5FFB CD F4 1F  1157              CALL   #PRINT
5FFE C9        1158              RET
5FFF           1159 IXYD
5FFF 00        1160              DB     $00
```

第107部 Small-C処理系の移植

●S-OS 6 周年記念

全機種共通システム最初のS-OS "MACE" をLISP-85とともにお届けして，早くも 7 年が過ぎました。S-OSシステムの成長を見守ってきたTHE SENTINELのコーナーも今回で84回，第107部を数えるまでにいたっています。

「どうして同じZ80というCPUを使っていながら，マシン語プログラムが機種ごとに違うのか」という素朴な疑問からスタートした試みは，いまや多くのアプリケーションプログラムを抱え，現役の 8 ビットOSとして，胸を張れるレベルにまで達したといえるのではないでしょうか。なかでもプログラミング言語シリーズの充実には目を見張るばかりです。プログラミング言語のアーキテクチャに影響を与えたエポックメイキングな言語の多くがユーザの手によって制作されています。ユーザの手によって進化するS-OSの面目躍如といったところでしょうか。

●S-OS初のCコンパイラ登場

その充実した言語シリーズのなかから抜けていたのがC言語です。お待たせいたしました。S-OS誕生当初からの要望であったC言語を，ようやくお届けすることができるようになりました。

思えば，コンパイラ自体が動いてから（オブジェクトを生成するわけではない），ずいぶん時間がたってしまいました。"SWORD" の環境にあった実行ファイルを生成するためにとリロケータブルアセンブラWZD，リンカWLK，ライブラリアンWLB……と，これまで皆さんに少しずつ用意してきていただいた環境が，これで一気に結実することになります。

先月バージョンアップし実数演算対応となったSLANG（REAL）ともども，今後のS-OS標準開発システムとしてご愛用ください。

とはいうものの，今回の掲載リストの入力だけならいざ知らず，移植元ディスクの入手もままならない人が多数いると思われます。そこでS-OS 6 周年記念として，特別に変更されたプログラムを限定配布します。メディアは 5 インチ2Dのみ。希望者はアンケートはがきにプレゼント番号 0 番を指定してください。投稿経験者優遇，あとは早い者勝ち＋抽選を原則としておきます（遠隔地は考慮します）。

なお，数に限りがありますので，なんらかのサークルに所属している人はどこかにサークル名も書いておいてください。こちらで重複を避ける（はずれるわけではない）ためにチェックします。

すみずみまで読んでいるといいことがあるという教訓ですね。なお，今後もこのような配布を行うと思ったら大間違いですのでその点は注意してください。

なお，予告されていたREALのソースリストは都合により来月以降に掲載となります。

●S-OSの系譜（21）

S-OS "MACE" が "SWORD" に進化したことの最大のメリットは，フロッピーディスクを扱えるようになった点ですが，同時にこれはフロッピーディスクの共通フォーマットができたということも意味します。S-OSで採用されたのはX1方式の記録方法で，これはPCシリーズやFMシリーズでも扱いやすい方式でした。つまり，S-OS"SWORD" はシャープのマシン以外のマシンへの移植性も考えて作られていたといえるでしょう。PC-8801，SMC-777，PASOPIAへとS-OSの世界は広がっていきました。

1987年 8 月号では，CPUに6809を採用するFM-7/77/AV用のS-OS "SWORD" が発表されています。6809でZ80のソフトウェアエミュレーションを行い，さらにその上でS-OSのシステムを動かすという手法で移植が行われました。さすがにスピードは100KHzのZ80程度だとコメントされてはいましたが，これはあくまで臨時用のもので，別売のZ80ボードを使用すれば，X1/MZシリーズと遜色ない環境が利用できるようになっていました。

さらに 8 月号では，S-OSで漢字を表示しようという試みがなされています。面白いのは漢字ROMを持たないマシンでも漢字表示できるように，漢字データをすべてディスク上に持っていることです。しかも，ビットイメージデータではなくストロークデータ。つまり，ベクタフォントが用意されたのです。フォントの表示にはMAGICを使います。グラフィックデータではなく，こういったデータを扱うというのも，MAGICの面白い使い方といえるでしょう。

ゲームとしては「碁石拾い」が発表されました。リアルタイムシューティング，テキアベが続いていたS-OSでは久々の思考型パズルゲームで，簡単な操作ながら楽しめるゲームに仕上がっていました。

全機種共通
システムS-OS"SWORD"要

処理系の移植
Small-C

Ishigami Tatsuya
石上 達也

お待たせしました。ついにS-OS上でC言語が稼働する日がやってきたのです。これで新しいS-OSの環境ができあがります。今回の作業はほとんどCP/M上で行いますので注意してください。

どうもお待たせしました。

予告で「Small-Cの移植」と出てからずいぶんと経ってしまいましたが，やっと発表できるようになりました。あの時点でもバージョン2.2は“SWORD”上で動いていたのですが「やっぱり，新バージョンじゃなきゃね」という，編集部のひと声によってそのプログラムは没となり，ここ3カ月ほど，バージョン2.7の移植に取り組んでいました。新バージョンといってもANSI準拠というわけではありません。とりあえず，Small-Cの1986年3月26日版です。

期待してくださった方々，お待たせしました。いよいよ，“SWORD”上でCコンパイラが動く日がやってきたのです。

このコンパイラの出現によって，“SWORD”上のプログラム開発環境は大幅に向上することを約束します。事実，私は先に発表したWZDやWLKで，このSmall-Cコンパイラにずいぶんと助けられています。皆さんのプログラム開発にぜひこのSmall-Cコンパイラを活用してください。

移植に際して必要なもの

・TPAが56Kバイト以上のCP/Mが動作する環境
・CP/M上で動作するリロケータブルアセンブラ（インテル，ザイログ両方のニーモニックがアセンブルできるもの）
・CP/M，“SWORD”ファイルコンバータ（Oh!X1989年8月号掲載）
・Small-C Ver. 2.7パッケージ（CUGライブラリNo.222,223）
・WZDシステム一式(WZD,WLK,WLB)（Oh!X1990年7,8,10月号にそれぞれ掲載）

Small-Cパッケージについて

JUG-CP/Mなどから直接CP/Mフォーマットで入手した方は注意が必要です。そのディスケットはあなたのパソコンとは違う機種のCP/Mを使って作られた可能性が大です。

たとえば，X1とPC-8001などのCP/Mでは，若干ファイルの構造が違うのです。詳しいことはここでは述べませんが，X1でPC用のCP/Mで書かれたディスケットをアクセスすると，読んだりするぶんには問題ありませんが，書いたり削ったりするのはマズイのです。この逆もいえます。

そこでまず，

A>PIP B：=A：*.*

とかやって，別のコピーを作ってください。ここで「なぁーんだ，バックアップを取っておけばよいのか」と思って，Format&Copy Utilityとか，CP/M上のCOPYコマンドを使わないでください。必ず，PIP.COMを使って転送してください。

それから，CP/M上で動作するリロケータブルアセンブラですがSLRでもMACRO-80でもかまいません。コムパックから出ているSmall-MACパッケージでもOKです。アセンブラファイルの8080→Z80コンバータがあれば（実は今回用に作ってあるんだな，これが），CUGライブラリの156番「Small-C with Float」というパッケージの中に収まっているZMAC.COMとZLINK.COMというのが使えそうです。

バージョン2.2と2.7の違い

では，Small-Cのバージョンが2.7になって，なにがよかったのでしょうか。

●まず，使えるデータのタイプがかなり増えた。たとえば，

char **argv

とか，

char *argv []

などのタイプが使えるようになった。

●3項演算子

expr1 ? expr2 : expr3

という構文が使えるようになった。

●includeレベルが6重まで許されるようになった。

●グローバル変数の2次元以上の配列が許されるようになった。

●16進数，8進数の定数が使えるようになった。

ライブラリのほうもかなり変更されていて，リダイレクト機能は貧弱になりましたが，そのぶん実行速度やメモリ効率はよくなりました。バージョン2.7の作者F.A. Scacchitti氏によると「UNIXライクな環境よりもCP/Mの環境にみあったライブラリ作りをこころがけた」のだそうです（と英語で書いてあった）。

コンパイラの移植

まず，コンパイラを移植します。この作業はCP/M上で行います。

最初にリスト１に示す変更を行ってください。この変更はコンパイラ本体に対する変更です。具体的には，

1)　パラメータの与え方を標準的にする(主にCC11.C,CC12.Cの変更)。

2)　出力されるアセンブラファイルをインテル形式からザイログ形式に変更する。そのついでに，Ｃで書かれていたCC4.C, CC41.C,CC42.CというファイルをハンドコンパイルしてSC4.MACというファイルにしておきました。これでコンパイルの速さがだいたい２倍になったはずです。

3)　#include分の閉じ括弧がない場合，暴走するというバグをとった。

＊　　＊　　＊

以上です。元のスタイルのほうがよいと思う方は1)の変更はいりません。

コンパイルの方法はバージョン2.7では少し変わっています(1)の変更で標準形式にしておきました)。アセンブラファイルはデフォルト状態では標準出力(画面のこと)に表示されてしまうのです。ここでは，リダイレクト機能を用いて以下のようにしてください。

```
A>CC CC1.C >CC1.MAC
-M -A -P -O -I
A>M80 =CC1
A>ERA CC1.MAC
A>CC CC2.C >CC2.MAC
-M -A -P -O -I
A>M80 =CC2
A>ERA CC2.MAC
A>CC CC3.C >CC3.MAC
-M -A -P -O -I
A>M80 =CC3
A>ERA CC3.MAC
A>M80 =SC4/Z
```

(SC4.RELを打ち込んだ方は結構です)

本体の変更が終わったら，次はライブラリ関係の移植を行います。変更の必要な関数はここですべて変更してもよいのですが，ここではコンパイラの移植に必要なものだけにしておきます（残りの関数は来月あたり“SWORD”上で移植します)。

まず，掲載されているダンプリストをすべて入力します。もちろん，ソースを入力してもかまいません。ソースで入力する場合，まずはリスト7のRDRTL.MACとリスト8のPOLL.MACを打ち込んでください。これらのファイルは変更というより，作り直しですので，元のファイルを読み込んで変更していくよりは，初めからエディタで打ち込んでいったほうがよいと思います。

これらのファイルのアセンブルが終わったらリンクを行います。

```
A>L80 /P:3000,RDRTL,CC1,
CC2,CC3,SC4,poll,clib/S,SC/E/N:P
```

このとき，リンカが実行開始アドレスはどこどこで，プログラムサイズはどのくらいであるというようなことを表示しますので，それをどこかにメモしておいてください。

ディスク上にSC.COMというファイルができあがっていることを確認してください。

このファイルが“SWORD”用のSmall-Cコンパイラなのですが，ファイルの先頭80Hバイトに無駄な部分があります。この部分は別にリンカのバグではなく，CP/Mから起動したときのために，転送ルーチンと実行開始アドレスに飛ばすプログラムが入っています。リスト5がこの部分をカットするためのプログラムです。

```
A>CUT SC.COM
```

として，SC.COMを生の純粋なバイナリファイルにします。

知っている人のために書いておきますと，HEXファイルを作っておいて，LOADコマンドで.COMファイルを作ってもよいのですが，2Dではすでにディスク容量が危機的状況にあります。2DDや2HDのディスク装置を持っている人ならばこの方法を使えます。

で，以上のようにして，できたSC.COMをファイルコンバータで“SWORD”上に転

虫繕い

●WZDのバグ

コマンドラインからRUNコマンドを拡張した“SWORD”を使用した場合，処理を終了して“SWORD”のモニタに戻ってくるときに誤動作することがありました。

```
3008  CD  AB  50  00
303A  CD  B2  50
3153  C3  FA  1F
347A  C3  FA  1F
50AC  5B  76  1F  13  13
50B1  C9  2A  76  1F  23  23  C9
```

以上の修正を加えてください。また，INC (IX+d) ,DEC (IX+d)のアセンブルがおかしかったので以下の修正をしてください。

```
6E52  C3  F3  35  00  00
6E7C  C3  F3  35  00  00
```

●WLKのバグ

一部，入出力ファイルが化けることがあった。

```
411A  21  00  00  CD  88  42
426E  2A  95  42  7E  2C  22  95  42
4276  37  3F  C0  21  E6  45  34  F5
427E  CD  97  42  38  02  F1  C9  F1
4286  37  C9  7D  32  95  42  7C  32
728E  E6  45  CD  97  42  C9  00
440E  CD  D8  43  2A  62  1F  16  00
4416  5F  19  36  8F  CD  C7  43  3E
441E  00  32  DD  42  C9
```

●WLBのバグ

.LIBファイルを読み込もうとすると動作がおかしくなることがあった。

```
345F  B8
364A  21  00  00  CD  B8  37
3725  C3  B1  39  00  00  00  00
379E  2A
37A1  7E  2C  22  C4  37  37  3F  C0
37A9  21  68  45  34  F5  CD  C6  37
37B1  38  02  F1  C9  F1  37  C9  7D
57B9  32  C4  37  7C  32  68  45  CD
57C1  C6  37
39B1  CD  14  39  2A  62  1F  16  00
39B9  5F  19  36  8F  CD  03  39  3E
39C1  00  32  19  38  C9
```

●MZ-80B/2000/2200用“SWORD”の不都合

SPがG-RAMのアドレスと重なっているときに #PEEK, #POKE をすると暴走する(電話で報告してくださった方ありがとうございました)。

対策です。下記のプログラムを実行してください。自動的に“SWORD”自身にパッチが当てられます。“SWORD”の拡張を行って1C00-1C24を使っている方はリスト9のソースリストを参考に各自，空いているエリアを使ってパッチを当ててください。

```
6000  01  21  00  11  31  15  21  1F
6008  60  ED  B0  01  24  00  11  00
6010  1C  21  3F  60  ED  B0  21  00
6018  1C  22  9B  1F  C3  FA  1F  E5
6020  F3  3E  01  D3  F7  DB  E8  CB
6028  FF  CB  B7  D3  E8  CB  FC  CB
6030  F4  7E  6F  DB  E8  CB  BF  CB
6038  F7  D3  E8  7D  FB  E1  C9  F5
6040  C5  E5  F3  47  3E  01  D3  F7
6048  DB  E8  CB  FF  CB  B7  D3  E8
6050  CB  FC  CB  F4  70  DB  E8  CB
6058  BF  CB  F7  D3  E8  FB  E1  C1
6060  F1  C9
```

(編注：S-OSではスタックは「自主管理」が原則です。WZDなどのようにプログラム側でスタックを#MEMAX付近に移したりすることはしないようにしてください)

送すればできあがりです。

以上の作業は思いっきりディスク容量を必要としますので，作業中に作成されるファイル(.BAKファイルなど)は用が済んだら消しておいて，ディスクスペースをなるべく多く確保しておいてください（アセンブルが終わったら一時的に作成される.ASMファイルなどもすぐに消してください)。

コンパイラの使い方

まず，ディスケットからコンパイラ本体を読み込みます。そして，実行アドレスを調べてそこに飛ばしてやればよいのです。"SWORD"の拡張を行っている方は，これらの動作は自動に行われます（近々，石上バージョンの拡張を行う予定です。ご期待ください)。

WZDやWLKのように，コマンドの後ろにパラメータを置くことができます（というか，パラメータを置かないと意味がない)。具体的にはコンパイラの実行アドレスが3000Hだったなら，

```
#LCC
#J3000 FILE -M -A -P -O
```

"SWORD"が拡張されていれば，

```
# CC： FILE -M -A -P -O
```

と入力することによって，FILE.CというC言語のソースファイルをコンパイルオプションM，A，P，O，Iでコンパイルすることができます（コンパイルオプションの意味は参考文献1のSmall-Cのマニュアルかパッケージについてくるドキュメントファイル（英語）を見てください)。

コンパイルが無事終了したなら，次はアセンブルですが，これは問題ないでしょう。

```
# WZD： =「ファイル名」
```

でOKです。

次にリンクですが，この際，自分で作ったファイルのほかに，ライブラリファイルが必要になってきます。たとえば，FILE1.RELとFILE2.RELというファイルをまとめてリンクする際，のちほど移植するclib.LIBというファイルが必要になってきます。実際のリンク方法は，

```
# WLK：
  *FILE1,FILE2
  *clib/S
  *FILE/N：P
```

でFILE.OBJという実行形式のファイルが得られます。

最後に

本当は高校2年生の頃（うーん，5年前か)からバージョン2.2のコンパイラ自体は動いていたのですが，いろいろありまして，発表が遅れてしまいました。Small-Cはソースが公開されていて解読するのも結構勉強になります。今回の移植作業が終了したら，ぜひ一度解読されることをおすすめします。

＊　＊　＊

最後に，今回の移植の機会を与えてくださったOh!X編集部とSmall-Cの原作者であるRon Cain，Jim Hexdrix，F.A.Scacchitti，およびライブラリの共著者であるL.E.Payneの各氏に感謝いたします。

参考文献

1)　DDJツールブック，DDJ編集部編/阿部尚子訳，工学社

2)　Small-C V2.7 for CP/M，F.A.Scacchitti，CUG library Disk No.222 & 223

Small-C の入手方法について

バージョン2.2だったら，コムパックからパッケージが発売されていたので，お持ちの方もいるかと思いますが，バージョン2.7の入手となるとパソコン通信をやっていない方にはちょっとつらいものがあります。

私はSmall-Cのパッケージを神田にあるVillage Centerというお店（PDS Houseというらしい。私はよく知らなかったが，ツウのあいだでは結構有名らしい）で買いました。ただし，ここに置いてあるCUG（C user's Group）のソフトは，たとえCP/M-80用であっても，PC-DOSの2Dフォーマットしかないそうです。私の某国民機RX21では読み込めましたが，X68000では，2Dは読めません。PC-9801などを使いPC-DOSフォーマットからCP/Mの2Dフォーマットになんとか変換してください。

で，PC-9801，神田まで行く暇/気力/金銭のない人は，JUG-CP/M（Japan User's Group of CP/M)というボランティアのPDSの配布団体が，CUGのソフトも取り扱っていますので，こちらから取り寄せてもらうといいでしょう。料金については1パッケージにつき会員1000円，非会員1500円だそうです。9枚までは，送料＝400＋枚数×200だそうです。詳しいことは，カタログについてくる「公開ソフト配布の手引き」を読んでください。

ただし，CUGのカタログディスクを見ると，こちらのSmall-Cのほうは圧縮してあるようです。解凍の方法は，きっと英語で書いてあるでしょう。

■Village Center
〒101
東京都千代田区神田神保町1-35-15　だるまビル1F
地下鉄　神保町駅から歩いて5分ぐらい。なかなか見つけづらいところにある。駅前の地図には「だるまビル」は載ってないので根気よく見つける。

■JUG-CP/M
〒112
東京都文京区大塚3-42-8　鈴木ビル201号
まず，270円分の切手を同封のうえここに，「カタログ希望」と書いた手紙を出す(TELとか，住所とか，名前とかは当然書く)。そのカタログ/入会案内/申込方法などをよく読んで，NO.222と223のディスクを申し込む。

中括弧について

S-OS "SWORD"には，中括弧"{"および"}"がサポートされていません。普通のASCIIコード表では"{"に$7B_H$，"}"に$7D_H$が割り当てられているのですが，"SWORD"では，$7B_H$にベタ塗りマークの"■"が，割り振られています。これを解除するには各"SWORD"内のexchgとかxechgとかのルーチンに手を加えてやればよいのですが，なかには機械自身が最初から'{'や'}'をサポートしていない機種もあります。そのときは以下のようにします。

まず，stdio.hに以下の2行を加えてやります。

```
#define BEGIN  {
#define END    }
```

これで，本来'{'を書くべきところに"begin"，'}'を書くべきところに"end"と書いてやればよくなります。たとえば，K&Rの最初のプログラムは，

```
main()
  begin
     printf("Hello World¥n");
  end
```

と，書けます。

さて，もともと'{'や'}'が使えないのに，どうやってこの2行をstdio.hに加えるのでしょうか？　それにはまずstdio.hに以下の2行を加えてやります。

```
#define BEGIN A
#define END   B
```

加えたら，いったんエディタを抜け出して，各機種用のマシン語モニタに入ります。そして，直接メモリエディットをして，41_Hを$7B_H$に，42_Hを$7D_H$に変更してやり，エディタをホットスタートして，stdio.hをセーブしてやればできあがりです。

▶浪人生はOh!Xがまぶしくて読めない。トホホ……。　長嶋　聡(18)栃木県

リスト1　変更点

```
==================  CC1.C  ==================
    24: /*nbflg,      no boot flag to return to CCP  (変更)*/
==================  CC11.C  ==================
/*
** get run options
*    (変更あり)
*/
ask()  {
  int i;
  i=listfp=nxtlab=0;
  output=stdout;

#ifdef OPTIMIZE
  optimize=
#endif /* OPTIMIZE */

  iflag=alarm=monitor=pause=NO;
  m80flg=YES;
  line=mline;

  while(++i < argcs) {
    line = argvs[i];
    if (line[0] == '-') {
      switch (toupper(line[1])) {
        case 'L':  if (isdigit(line[2]) & (line[3] <= ' ')) {
                     listfp = line[2] - '0';
                     break;
                     }
                   else
                     goto errcase;
        case 'A':  alarm = YES;
                   break;
        case 'M':  monitor = YES;
                   break;
        case 'P':  pause = YES;
                   break;
        case 'I':  iflag = YES;
                   break;
#ifdef OPTIMIZE
        case 'O':  optimize = YES;
                   break;
#endif /* OPTIMIZE */

#ifndef LINK
        case 'B': if (numeric(line[2]) & (line[3] <= ' ')) {
                    bump(0); bump(2);
                    if(number(&nxtlab)) break;
                    }
                    /* fall through to error case */
#endif /* LINK */

        errcase:
    default:  sout("usage: cc [file]...[-a] [-i] [-l#] [-m] [-n] [-p]", stderr);

#ifdef OPTIMIZE
                   sout(" [-o]", stderr);
#endif /* OPTIMIZE */

#ifndef LINK
                   sout(" [-b#]", stderr);
#endif /* LINK */

                   sout("¥n", stderr);
                   abort(ERRCODE);
          }
        }
      }
    }


/*
** input and output file opens (変更あり)
*/
openfile() {          /* entire function revised *//*39*/
  char fn[20];            /* file name buffer 2 + 13 + 1 + 3 + 1 */
  char *cmdptr;           /* command line pointor */
  int i, ext;
  input=EOF;
  line = pline;

  while(++filearg < argcs) {
        cmdptr = argvs[filearg];
        if(cmdptr[0]=='-') continue;

        ext = NO;
        i = 0;

        while(cmdptr[i] && i < 15) {
                if(cmdptr[i] == '.') {
                        ext = YES;
                        break;
                }
                fn[i] = cmdptr[i++];
        }
        if(!ext) strcpy(fn + i, ".C");

        input = mustopen(fn, "r");

        if(!files && isatty(stdout)) {
                strcpy(fn + i, ".ASM");
                output = mustopen(fn, "w");
        }
        files=YES;
        kill();
        return;
  }
  if(files++) eof=YES;
  else input=stdin;
  kill();
  }

/*
** open a file with error checking (quoted from ver 2.2) (変更あり)
*/
mustopen(fn, mode) char *fn, *mode; {          /*39*/
  int fd;
  if(fd = fopen(fn, mode)) return fd;
  sout("open error on ", stderr);
  lout(fn, stderr);
  abort(ERRCODE);
  }


==================  CC12.C  ==================
/*
** open an include file
** (変更あり)
*/
doinclude()  {
  char c, *fname, buff[17];
  int i;

  blanks();       /* skip over to name */
  /*
   * added code to handle include filename in quotes or brackets
   * 4/5/83 br
   */
  if ((*lptr == '"') | (*lptr == '<')) {
    fname = buff;
    for(i = 0; i < 16; i++) {
      c = *lptr++;
      if ((c == '"') | (c == '>')) break;
      *fname++ = c;
      }
    *fname = '¥0';
    fname = buff;
    }
  else
    fname = lptr;       /* no '"' or '<' (original convention) */

   if(inclevel <= 5){                                      /* fas 2.7 */
     if((input2[++inclevel]=fopen(fname,"r"))==NULL) {     /* fas 2.7 */
       input2[inclevel--]= EOF;                            /* fas 2.7 */
       error("open failure on include file");              /* fas 2.7 */
       }                                                   /* fas 2.7 */
   }else{                                                  /* fas 2.7 */
     error("maximum include nesting reached");             /* fas 2.7 */
   }                                                       /* fas 2.7 */

  kill();         /* clear rest of line */
      /* so next read will come from */
      /* new file (if open) */
  }

==================  cc21.c  ==================
   218:     if (input == EOF) openfile();       /* (変更) */

/*
 * special version of 'fgets' that deletes trailing '¥n'
**              (変更あり)
 */
xgets(string, len, fd) char *string; int len, fd; {

  char c, *strptr;
  strptr = string;
  while (((c = getc(fd)) != '¥n') && (--len)) {
    if (c == EOF) return NULL;
    if (c == 8){                                 /* fas 2.2 */
      len++;
      string--;
    }else{
      *string++ = c;                             /* no mask parity off */
      }
    }
  *string = NULL;
  return strptr;
  }
```

リスト2　CC4.REL

```
0000   84 D4 D0 CD 20 64 84 54   : 51
0008   14 44 55 28 19 51 49 05   : 8D
0010   25 31 16 06 45 58 54 45   : A8
0018   52 4E 81 93 13 D0 51 10   : F8
0020   54 A0 64 F5 55 44 84 55   : BF
0028   88 15 15 39 51 49 66 06   : F1
0030   49 4E 44 49 52 45 81 91   : CD
0038   91 90 D0 53 13 20 64 74   : 4F
0040   55 44 D4 54 D8 19 1D 15   : E4
0048   51 31 3D 0E 06 50 55 54   : CC
0050   4D 45 4D 81 94 15 55 14   : 72
0058   D5 12 E0 44 D4 F5 64 58   : 90
0060   11 4D 5D 05 42 05 53 57   : B1
0068   41 50 32 81 52 53 53 51   : 8D
0070   51 20 64 94 D4 D4 54 43   : A8
0078   28 11 41 55 4D 22 06 53   : 97
-------------------------------------
SUM:   58 C4 BB EE 97 90 6C 21   76FE

0080   4D 41 52 54 50 80 D4 13   : EB
0088   D4 20 65 35 74 15 05 35   : 51
0090   48 09 4D 5E 05 46 46 52   : DF
0098   45 54 81 90 D0 53 13 14   : F4
00A0   D5 20 45 54 C5 43 08 11   : AF
00A8   29 55 35 42 06 50 52 49   : E6
00B0   4E 54 4C 81 96 91 54 93   : 7D
00B8   D2 95 60 64 44 54 65 35   : 5D
00C0   44 F8 15 41 3D 25 39 52   : 7F
00C8   06 4D 4F 44 53 54 4B 81   : 59
00D0   91 13 D5 50 93 11 60 54   : 21
00D8   64 64 14 44 48 15 19 19   : AF
00E0   4D 55 0A 06 46 46 4D 55   : E0
00E8   4C 54 81 51 91 91 12 55   : FB
00F0   A0 54 64 64 D4 F4 48 11   : DD
00F8   19 19 3D 4A 05 46 46 58   : A2
-------------------------------------
SUM:   5D EE 24 10 59 56 2F 23   6E3C

0100   4F 52 81 51 91 90 53 91   : 78
0108   20 44 C4 E4 54 78 15 19   : 06
0110   19 05 4D 4A 05 46 46 41   : 87
0118   53 4C 80 D3 91 51 E0 34   : E8
0120   34 F4 D8 15 1D 15 51 35   : CD
0128   12 03 49 4E 43 80 D1 11   : 51
0130   50 E0 44 64 64 55 18 0D   : B6
0138   15 44 C2 04 46 46 4E 45   : 3E
0140   81 95 11 54 D5 12 95 60   : 57
0148   34 E4 53 08 11 19 19 31   : E7
0150   52 03 4C 54 30 81 11 91   : 48
0158   93 11 60 34 C4 53 08 11   : 68
0160   19 19 1D 52 03 47 54 30   : 6F
0168   81 11 91 91 D1 60 34 74   : 8D
0170   53 08 0D 55 31 52 03 55   : 98
0178   4C 45 80 D5 51 D5 20 35   : 61
-------------------------------------
SUM:   59 06 84 0E B5 9C 88 18   4867

0180   54 74 58 19 41 15 15 41   : E5
0188   21 3E 06 50 4F 53 54 4C   : F7
0190   41 94 00 00 4D 6E C3 19   : 6C
0198   A0 00 00 88 00 00 64 8A   : 16
01A0   80 00 00 89 DC 02 19 11   : 11
01A8   00 00 02 AA 10 00 11 57   : 24
01B0   03 83 2E D2 DD 0F 00 53   : C5
01B8   BB 4A 4D AA 8E 00 2A A3   : 57
01C0   40 07 E7 F0 10 40 11 11   : 90
01C8   88 CF CF E0 20 80 14 2A   : E4
01D0   A4 60 01 10 40 00 32 CD   : 54
01D8   BD 80 02 AA 64 00 11 08   : 66
01E0   80 03 22 2A 78 00 18 65   : C4
01E8   08 6B 40 03 95 9A 00 00   : E5
01F0   30 4F 8B 51 40 3C 22 06   : FF
01F8   00 06 4F CF E0 20 80 0C   : B0
-------------------------------------
SUM:   75 8C D0 77 35 9D 06 15   335B

0200   21 AD A0 0C DB D8 00 21   : 4E
0208   AE 60 0C 3B F2 16 6D 30   : FA
0210   9A 4D C0 02 91 10 A4 54   : 42
0218   26 00 04 04 52 71 10 00   : 01
0220   21 B8 40 0E 51 50 00 00   : C8
```

▶「SION」は面白い！　だけどキーボードでゲームをするときには，普通Z,X,Cキーを使うのにどうしてひとつずれてX,C,Vなんだろう。XCV?!　エックスシーブィ?!　まさか，エクシヴィをもじっているのですか？
柳川　史彦(18)茨城県

```
0228  E5 10 82 C0 0E 51 55 7D  : 68
0230  00 08 A5 80 A1 97 28 85  : 12
0238  04 00 72 88 40 00 07 2B  : 70
0240  34 00 00 75 88 41 80 01  : F3
0248  CB E5 D6 7C 5A B0 18 2E  : 52
0250  17 2B 14 F8 B5 65 5B C0  : 83
0258  01 0D CB 00 66 D4 00 C6  : D9
0260  0B 85 CA C5 72 B3 6B 20  : CF
0268  63 05 86 00 00 08 6E 90  : F4
0270  03 0E FC 85 93 89 E4 32  : C4
0278  19 04 A4 47 21 80 04 04  : B1
-----------------------------------
SUM:  3A E3 EE 9D 13 95 59 6D  B925

0280  C2 20 80 82 2C 00 08 0B  : 23
0288  04 F2 90 80 82 00 10 80  : 18
0290  00 0C DA 80 18 CD 00 00  : 4B
0298  07 40 00 02 9D 89 78 00  : E7
02A0  18 76 E8 01 CA 21 BE E0  : 00
02A8  0C DA 80 18 E1 61 DF 90  : 2F
02B0  B1 01 14 B0 54 10 00 18  : F2
02B8  2E 17 2B 14 45 5A 00 4B  : 6E
02C0  C2 32 B0 84 06 00 0C 93  : CD
02C8  81 80 D6 55 41 01 10 80  : FE
02D0  80 01 93 F3 F8 04 20 00  : 23
02D8  83 04 20 00 01 F8 68 B9  : C1
02E0  1C 04 C3 4F E0 18 80 1C  : C6
02E8  21 02 00 03 27 E1 55 68  : EB
02F0  01 11 88 CE E2 31 B0 00  : 2B
02F8  75 47 00 57 23 80 28 F8  : D6
-----------------------------------
SUM:  C9 DB 15 A4 F3 E9 7E A6  9E1F

0300  02 32 00 00 05 55 A0 04  : 32
0308  46 23 3F 3F 80 82 1A AA  : AD
0310  02 CA BB 00 62 1A BE 02  : C3
0318  C3 BB 00 64 32 19 1C 86  : CF
0320  48 20 94 84 62 30 01 0C  : 1F
0328  86 47 24 93 8A 82 31 18  : D9
0330  02 5A D4 02 5E 11 95 83  : B9
0338  4F E0 18 A0 48 21 00 80  : D0
0340  03 27 E7 F0 10 40 1B 10  : 7C
0348  DA B0 16 AB 36 A0 06 34  : 5B
0350  44 20 80 00 65 9B 50 03  : 37
0358  04 36 CC 05 9B 7E 42 C4  : 2A
0360  36 D4 05 87 76 00 C4 36  : 06
0368  F4 05 AA CD A8 01 8D 11  : B7
0370  08 20 00 19 66 D4 00 C1  : 3C
0378  0D C6 01 61 DF 90 B1 01  : 56
-----------------------------------
SUM:  90 67 97 CA 54 4C 10 71  C217

0380  30 88 20 20 8B 05 00 01  : 89
0388  48 00 86 43 29 96 0A 82  : 5C
0390  31 18 00 40 4C 22 08 09  : 08
0398  04 C1 60 A0 00 29 00 04  : F2
03A0  40 C0 00 C9 F8 46 66 37  : A4
03A8  BB 4B 60 00 02 9D DA 52  : 31
03B0  72 B3 40 00 06 08 86 F1  : EA
03B8  00 B0 EF C8 58 80 82 44  : 05
03C0  22 08 09 04 C1 61 4C A0  : 45
03C8  00 60 B8 5C AC 52 F0 8C  : EE
03D0  AC 1A 7F 00 C5 01 E1 08  : F4
03D8  04 00 19 3F 3F 80 82 00  : 9D
03E0  A8 86 96 01 35 59 B5 00  : 08
03E8  31 A2 21 04 00 03 2C DA  : 01
03F0  80 18 21 A7 40 4C 3B F2  : 19
03F8  16 21 A7 C0 4D 56 6D 40  : EE
-----------------------------------
SUM:  5B B2 6D DF 8B 83 82 8E  8024

0400  0C 68 88 41 00 00 CB 36  : 3E
0408  A0 06 08 6A 20 13 0E FC  : 55
0410  85 88 09 84 41 01 04 58  : 38
0418  4C 06 88 09 84 41 00 A0  : 48
0420  00 29 16 10 40 02 02 61  : F4
0428  10 40 28 00 0A 45 84 82  : CD
0430  60 00 45 5A 00 47 54 FC  : 96
0438  05 4E CA AC A0 45 E1 19  : A8
0440  58 46 23 09 84 C2 61 30  : A1
0448  D1 DC 46 36 00 2B C6 55  : 6F
0450  34 08 55 5A 00 44 62 33  : C4
0458  F3 F8 08 21 AF 20 4C AB  : DA
0460  F2 16 21 B1 80 4C 3B B0  : 91
0468  06 20 26 11 04 04 11 61  : D7
0470  30 1A 20 26 11 04 02 82  : 29
0478  21 14 52 2C 20 80 08 64  : BF
-----------------------------------
SUM:  8B 39 ED 1C B7 4D C3 7C  06AF

0480  32 81 24 9C 54 11 88 C0  : 20
0488  02 1B 36 04 C3 BF 21 62  : 5C
0490  02 61 10 40 44 16 12 01  : 20
0498  A2 02 61 10 40 45 16 13  : C3
04A0  00 02 1B 62 04 C3 BF 21  : 26
04A8  62 02 29 60 40 44 22 8B  : 1E
04B0  09 04 C0 00 86 E0 81 30  : E4
04B8  EA 00 60 80 98 44 10 12  : C8
04C0  09 82 C0 00 86 E7 81 30  : 69
04C8  EA 00 60 80 98 44 10 11  : C7
04D0  08 A2 C0 00 86 F3 01 33  : 17
04D8  6F C8 58 AA E9 00 8A C5  : 71
04E0  62 2B BC 04 C9 10 14 0A  : 44
04E8  A5 32 40 80 90 4C 1D 80  : 10
04F0  18 2D 17 0B 95 AA C5 76  : E1
04F8  94 E8 70 18 44 14 00 0C  : 68
-----------------------------------
SUM:  4A 65 EA 03 BC 8E 55 69  75E7

0500  9F 84 66 63 7A D1 85 67  : 23
0508  80 DD A5 BA 1C 06 CD A2  : 4D
0510  00 6C 99 68 80 18 86 BE  : 49
0518  81 9F 94 E2 80 28 48 46  : CC
0520  13 0C 3D DD A5 3B 24 06  : 43
0528  2A 00 00 04 56 50 0D DA  : BB
0530  5B B2 80 61 B7 69 4E CA  : 26
0538  01 8A AC 90 1A 9D DA 52  : AA
0540  18 0F 45 56 50 0C 23 61  : A2
0548  C0 D9 B7 C0 2C 40 E4 15  : 75
0550  59 40 31 58 8A CA 01 8F  : 06
0558  80 43 2B 2C 06 2A B2 80  : 7C
0560  62 B1 15 94 03 3F 33 6C  : 9D
0568  B8 18 E1 96 3A B2 C0 64  : 57
0570  73 F2 40 EE FE 18 0C 00  : B5
0578  EC 60 51 DC 7C 01 0C 00  : 02
-----------------------------------
SUM:  63 3A 80 C7 25 F2 3E 5E  BF71

0580  55 E3 2B 2C 06 C3 A9 C0  : C1
0588  62 AB C6 04 23 11 88 AB  : 3E
0590  A8 1B 24 40 45 2C 08 08  : A8
0598  84 51 61 20 98 00 10 10  : 0E
05A0  48 84 41 01 20 98 2C 29  : 1B
05A8  94 00 12 D9 40 39 6C B0  : 14
05B0  1C B6 5C 0D FC 30 6C 3F  : 12
05B8  87 43 F6 48 86 D6 81 B3  : 98
05C0  6F C8 58 AA BD 01 88 C4  : 43
05C8  62 2B 4A 06 C9 10 14 09  : D3
05D0  E5 01 01 10 8A 00 10 DB  : 6C
05D8  B0 36 1D F9 0B 10 11 4B  : 73
05E0  02 01 41 4C A0 29 16 12  : 81
05E8  09 80 01 0D CD 03 61 DD  : A5
05F0  80 32 19 0C A6 57 24 95  : 8D
05F8  08 64 81 18 8C 00 E5 10  : 86
-----------------------------------
SUM:  5B B8 B7 F5 A2 7B 0B D5  95E2

0600  DE 30 36 6D 40 0C 70 B0  : 1D
0608  EF C8 58 80 86 41 26 13  : 8F
0610  04 00 01 0D F0 03 61 DF  : 45
0618  90 B1 01 48 8A 54 00 08  : 70
0620  6F D8 1B 0E EC 01 90 C8  : B5
0628  64 42 19 04 98 23 11 80  : 0F
0630  04 34 50 11 9B 50 03 18  : 9F
0638  2E 17 2B 15 9B 41 43 18  : BC
0640  77 A8 00 40 4A 28 08 00  : D9
0648  0C 17 0B 95 8A 11 07 00  : 65
0650  03 2C D0 00 01 CA 21 00  : EB
0658  00 1C AC D0 00 01 82 C1  : DC
0660  10 81 80 03 96 6D 22 04  : 3D
0668  72 88 40 C0 01 CB 36 99  : 95
0670  82 38 D9 A0 00 03 05 92  : CD
0678  11 00 80 14 EE D2 90 86  : 7B
-----------------------------------
SUM:  01 56 DF 96 54 6A 7D 98  C252

0680  AA 82 32 AA 00 60 86 AD  : 9B
0688  02 30 EA 00 60 80 88 42  : C6
0690  10 00 04 04 42 B8 80 00  : 92
0698  21 AC A0 8C 3B F2 16 20  : 5C
06A0  22 15 C4 02 41 58 C8 00  : 5E
06A8  C1 68 B8 5C AD 56 28 8A  : F2
06B0  A7 82 BB 4A 75 44 15 DA  : D6
06B8  5B B5 40 6A 77 69 48 45  : 27
06C0  54 C1 4F 8B 56 55 15 05  : B4
06C8  57 AD 1F 56 F4 10 22 07  : A6
06D0  00 29 DD A5 21 80 9C 75  : 5D
06D8  54 C1 5C C0 11 40 28 43  : ED
06E0  46 41 59 B7 E4 2C 43 54  : 3E
06E8  C1 46 A3 AA A6 0A A7 14  : BF
06F0  1A 04 34 84 15 9B 7E 42  : 46
06F8  C7 55 4C 14 7A 3D 19 55  : A1
-----------------------------------
SUM:  A9 4A 5A 8B 4C 18 6D 7B  3E24

0700  30 50 C3 A8 55 54 C1 55  : AA
0708  EB 47 95 F7 04 08 BE 5F  : E7
0710  EA 77 69 48 70 2A 1D 55  : 1E
0718  30 57 30 04 50 0D 10 D2  : FA
0720  90 56 6D F9 0B 1D 55 30  : F9
0728  51 E0 CA A9 82 8E AA 98  : F6
0730  2A 9C 50 2E 10 D3 10 56  : 8D
0738  6D F9 0B 1D 55 30 51 E0  : 44
0740  F0 65 54 C1 43 0E A1 55  : B1
0748  51 05 3E 2D 58 96 AC 08  : 63
0750  55 54 C1 5C AC DB B2 02  : 01
0758  C1 10 D3 A0 56 6D F9 0B  : 0B
0760  15 55 10 53 E2 D5 89 6A  : 77
0768  C0 85 55 3C 15 92 20 24  : C1
0770  93 88 62 02 99 40 00 20  : 78
0778  28 13 CA 02 02 11 0C 00  : 26
-----------------------------------
SUM:  94 73 3A 55 3A E5 B9 F1  29FD

0780  20 22 11 48 62 02 99 40  : D8
0788  00 20 28 15 4A 64 81 01  : 8D
0790  08 86 00 10 10 48 84 41  : BB
0798  01 20 98 2C 29 94 01 A2  : 45
07A0  02 61 10 40 53 28 0B 09  : 42
07A8  04 C0 04 B5 44 16 5A A6  : D7
07B0  0B 2D 55 05 10 D5 B0 56  : 7D
07B8  1D F9 0B 10 10 48 84 41  : 4E
07C0  01 20 98 2C 24 13 00 02  : 1E
07C8  1A D8 0A C3 BF 21 62 02  : 03
07D0  09 10 88 20 24 13 05 84  : 81
07D8  42 28 00 43 5F 41 58 77  : 1C
07E0  60 0C 86 43 29 95 48 42  : 7D
07E8  31 18 00 43 62 C1 58 77  : 7E
07F0  60 0C 86 43 26 95 49 85  : BE
07F8  41 18 8C 00 21 B3 40 AC  : A5
-----------------------------------
SUM:  EF A7 07 BE D4 C3 20 53  C6E1

0800  3B B0 06 43 21 91 09 25  : 14
0808  61 18 8C 00 CD B2 80 AC  : B0
0810  3B 56 04 21 B5 C0 AC 3B  : 12
0818  B0 06 43 21 93 CA 42 31  : EA
0820  18 00 43 6E C1 58 77 60  : B9
0828  0C 86 43 2C 13 CA 42 31  : 51
0830  18 00 43 72 41 58 77 60  : 3D
0838  0C 86 43 20 93 88 82 31  : C3
0840  18 00 43 75 C1 58 77 60  : C0
0848  0C 86 43 26 13 88 A4 71  : AB
0850  18 8C 00 21 BC C0 AC 3B  : 28
0858  B0 06 43 21 90 4A 65 21  : 7A
0860  18 8C 00 21 BE 80 AC 3B  : EA
0868  B0 06 43 21 90 4A 64 C1  : 19
0870  18 8C 00 21 A0 40 CC 3B  : AC
0878  B0 06 43 21 93 88 A4 71  : 4A
-----------------------------------
SUM:  4B 6C 34 12 7F 4B D5 34  3DD3

0880  18 8C 00 21 A2 00 CC 3B  : 6E
0888  B0 06 43 21 90 C9 E4 D1  : 28
0890  18 8C 00 CD BA E0 4C DB  : 32
0898  76 04 C1 70 B9 58 AE 56  : C0
08A0  68 00 00 C1 66 D1 20 46  : C6
08A8  6D 85 05 61 D9 E0 36 0B  : 52
08B0  85 CA C5 72 88 6A 28 33  : D3
08B8  36 FC 85 B8 4F 8B 56 40  : DF
08C0  AC F8 B5 10 3C 19 22 02  : E2
08C8  49 38 86 20 24 13 00 0C  : 6A
08D0  17 0B 95 8A E5 10 D6 20  : 2C
08D8  66 6D F9 0B 70 9F 16 AC  : A8
08E0  81 59 F1 6A 20 78 32 44  : 43
08E8  04 42 29 0C 40 48 26 00  : 29
08F0  04 35 C0 19 87 76 00 C8  : D7
08F8  64 32 29 44 46 23 00 08  : 74
-----------------------------------
SUM:  45 17 1F 63 9D DB E4 EF  9FDD

0900  6C 38 33 36 A0 06 30 5C  : 3F
0908  2E 56 2B 36 82 86 30 E9  : 06
0910  40 30 80 98 44 10 10 45  : 31
0918  84 80 68 80 9E 52 10 13  : FF
0920  01 A2 02 51 40 40 4E 2D  : F1
0928  0B 00 02 1B 46 0C C3 BB  : F8
0930  00 64 32 19 38 8A 23 11  : A5
0938  80 04 36 E8 19 9B 50 03  : A9
0940  18 2E 17 2B 15 9B 41 43  : BC
0948  18 76 14 18 40 4C 22 08  : 70
0950  08 22 C2 40 34 40 4F 29  : 18
0958  08 09 80 D1 01 28 A0 20  : 4B
0960  2D 0B 00 02 1B AA 0C C3  : CE
0968  BB 00 64 32 19 30 A8 23  : 65
0970  11 80 04 37 B0 19 9B 50  : 80
0978  03 18 2E 17 2B 15 9B 41  : 7C
-----------------------------------
SUM:  26 BA B5 C7 74 B6 40 A4  D27B

0980  43 18 76 E0 18 40 58 27  : 88
0988  94 84 04 10 68 80 9E 52  : 04
0990  10 12 01 A2 02 51 40 40  : 98
0998  50 16 00 04 34 18 1D 87  : 5A
09A0  76 00 C8 64 32 61 14 46  : 8F
09A8  23 00 08 68 E8 3B 36 A0  : 8C
09B0  06 30 5C 2E 56 2B 36 82  : F9
09B8  86 30 EF 50 30 80 98 44  : 81
09C0  10 10 45 84 80 68 80 9E  : EF
09C8  52 10 13 01 A2 02 51 48  : B3
09D0  40 5A 16 09 05 63 80 68  : 09
09D8  80 B0 4F 29 08 08 20 D1  : A9
09E0  01 3C A4 20 24 03 44 04  : 70
09E8  A2 80 80 A0 2C 00 08 6A  : E0
09F0  78 3B 0E EC 01 90 C8 64  : 6A
09F8  72 A0 8C 46 00 10 D7 60  : 2B
-----------------------------------
SUM:  0B E5 11 89 D6 E8 C7 3D  7B6D

0A00  76 6D 40 0C 60 B8 5C AC  : 4F
0A08  56 6D 05 0C 66 D1 B0 71  : 2C
0A10  0D 8A 07 66 D4 00 C6 0B  : A9
0A18  85 CA C5 66 D0 50 C6 1D  : 7D
0A20  64 07 10 16 09 E5 21 01  : A1
0A28  04 1A 20 27 94 84 04 80  : 01
0A30  68 80 94 50 10 13 45 80  : B4
0A38  01 01 3C A4 20 26 03 44  : 6F
0A40  04 A2 80 80 B4 2C 00 08  : 8E
0A48  6C E8 3B 0E EC 01 90 C8  : E2
0A50  64 72 28 8C 46 00 10 DB  : BB
0A58  40 76 6D 40 0C 60 B8 5C  : E3
0A60  AC 56 6D 05 0C 61 D7 40  : F8
0A68  71 01 60 9E 52 10 10 41  : 23
0A70  A2 02 79 48 40 48 06 88  : 7B
0A78  09 45 01 01 34 58 00 10  : EC
-----------------------------------
SUM:  0B E0 A8 5B FB 19 4A AA  121B

0A80  DC E0 76 1D D8 03 21 90  : DB
0A88  CA A4 C2 A0 8C 46 00 10  : B2
0A90  DD C0 76 1D D8 03 21 90  : BC
0A98  CA A4 C2 28 8C 46 00 10  : 3A
0AA0  DE A0 76 1D D8 03 21 90  : 9D
0AA8  CA A4 72 A0 8C 46 00 10  : 62
0AB0  DF 80 76 1D D8 03 21 90  : 7E
0AB8  CA A4 72 28 8C 46 00 11  : EB
0AC0  5E 80 B1 55 E8 0B 3F 29  : 3F
0AC8  D9 0C DA 4A 10 38 7A 8A  : 55
0AD0  AF 40 59 F8 46 22 BD 01  : 66
0AD8  62 60 01 BD CA 2A 00 00  : 74
0AE0  1C AC D0 00 01 82 C1 0C  : E8
0AE8  37 82 35 5C 20 0C 04 1D  : 97
0AF0  00 00 0A 71 40 08 0C 0F  : DE
0AF8  15 5E 80 B6 AB 36 F8 05  : 87
-----------------------------------
SUM:  4E 08 B4 DB A4 7F C3 72  65F9

0B00  B4 45 00 E7 5B 36 F5 05  : 6B
0B08  88 D9 B7 A8 2C 46 EB 08  : 25
0B10  3A 94 EC 91 15 E8 0B 75  : C8
0B18  B3 6F 50 58 8D 9B 7E 42  : B2
0B20  C6 EC 91 01 30 88 20 24  : 40
0B28  13 05 83 00 68 80 82 44  : 49
0B30  22 08 09 04 C1 61 4C A0  : 45
0B38  0D 10 10 C8 24 C2 60 80  : BB
0B40  86 43 23 92 49 C5 41 18  : E5
0B48  8C 1A 20 22 96 04 04 42  : C8
0B50  28 B0 90 4C 06 80 04 05  : 43
0B58  02 79 40 40 44 22 83 44  : 28
0B60  05 02 A9 4C 90 20 22 11  : DF
0B68  40 02 02 61 10 40 48 26  : 63
```

▶「SION」のパワーアップアイテムがSとMなんてやりますねえ。谷　聡雄(18)茨城県

```
0B70  0B 06 00 D1 01 04 88 44  : B3
0B78  10 12 09 82 C2 99 40 1A  : 62
-----------------------------------
SUM:  CD CC E7 85 32 92 B5 84  7276

0B80  20 21 90 49 84 C1 01 0C  : 6C
0B88  86 47 24 93 8A 82 31 18  : D9
0B90  34 00 20 28 13 CA 02 02  : 5D
0B98  41 30 1A 20 28 15 4A 64  : 96
0BA0  81 01 20 98 00 10 13 08  : 65
0BA8  82 02 41 30 58 32 06 88  : 0D
0BB0  08 24 42 20 80 90 4C 16  : 00
0BB8  14 CA 00 D1 01 0C 82 4C  : 8A
0BC0  26 08 08 64 32 39 24 9C  : C5
0BC8  54 11 88 C1 A2 02 29 60  : DB
0BD0  40 44 22 8B 09 04 C0 68  : 66
0BD8  00 40 50 27 94 04 04 22  : 75
0BE0  18 34 40 50 27 94 04 04  : 9F
0BE8  42 28 34 40 50 2A 94 C9  : B5
0BF0  02 02 21 14 1A 20 28 15  : B0
0BF8  4A 64 81 01 08 86 00 10  : CE
-----------------------------------
SUM:  9A E8 A9 59 2C A7 36 F4  E90E

0C00  13 08 82 02 41 30 58 32  : 9A
0C08  06 88 08 24 42 20 80 90  : 2C
0C10  4C 16 14 CA 00 D1 01 0C  : 1E
0C18  82 4C 26 08 08 64 32 39  : D3
0C20  24 9C 54 11 88 C1 A0 01  : 0F
0C28  01 40 9E 50 10 10 88 60  : 37
0C30  D1 01 40 9E 50 10 12 09  : 2B
0C38  80 D1 01 40 AA 53 24 08  : BB
0C40  09 04 C0 68 80 A0 55 29  : D3
0C48  92 04 04 22 18 00 40 41  : 55
0C50  22 11 04 04 82 60 B0 A6  : 73
0C58  50 06 88 08 64 12 61 30  : ED
0C60  40 43 21 91 C9 24 E2 A0  : A4
0C68  8C 46 0D 00 08 08 64 12  : 65
0C70  61 30 40 43 21 91 0A 64  : 34
0C78  72 48 8C 46 00 10 10 48  : F4
-----------------------------------
SUM:  09 C0 41 E7 8D 98 6F 17  F4EB

0C80  84 41 01 20 98 2C 22 11  : DD
0C88  41 A2 02 19 04 98 4C 10  : F6
0C90  10 C8 64 72 49 38 A8 23  : FA
0C98  11 83 40 02 02 19 04 98  : 8D
0CA0  4C 10 10 C8 64 42 21 1C  : 17
0CA8  92 23 11 80 04 04 12 21  : 81
0CB0  10 40 48 26 0B 0A 65 00  : 38
0CB8  68 80 86 41 26 13 04 04  : F0
0CC0  32 19 1C 86 48 20 94 84  : 6D
0CC8  62 30 68 00 40 43 20 93  : 30
0CD0  09 82 02 19 0C 88 53 23  : B0
0CD8  90 C4 62 30 00 80 82 44  : 2C
0CE0  22 08 09 04 C1 61 10 8A  : F3
0CE8  0D 10 10 C8 24 C2 60 80  : BB
0CF0  86 43 23 90 C9 04 12 90  : EB
0CF8  8C 46 0D 00 08 08 64 12  : 65
-----------------------------------
SUM:  AA 51 C7 87 CA 12 25 47  AAE4

0D00  61 30 40 43 21 91 08 84  : 52
0D08  72 18 8C 46 00 10 10 48  : C4
0D10  84 41 01 20 98 2C 29 94  : 67
0D18  01 A2 02 61 10 40 44 16  : B0
0D20  12 01 A2 02 61 10 40 45  : AD
0D28  16 13 01 A2 02 19 04 98  : 83
0D30  4C 10 10 C8 64 72 49 38  : 8B
0D38  A8 23 11 83 44 04 92 71  : AA
0D40  0C 40 48 26 03 44 04 32  : 37
0D48  09 30 98 20 21 90 CA 04  : 70
0D50  92 71 50 46 23 06 80 04  : 46
0D58  04 32 09 30 98 20 21 90  : D8
0D60  C9 24 E2 19 24 46 23 00  : 75
0D68  08 08 24 42 20 80 90 4C  : F2
0D70  16 14 CA 00 D1 01 30 88  : 7E
0D78  20 22 0B 09 00 D1 01 30  : 58
-----------------------------------
SUM:  26 E7 A7 19 C8 3E F7 CA  B3E3

0D80  88 20 22 8B 09 80 D1 01  : B0
0D88  0C 82 4C 26 08 08 64 32  : A6
0D90  39 24 9C 54 11 88 C1 A2  : 49
0D98  02 21 14 86 20 24 13 01  : 15
0DA0  A2 02 19 04 98 4C 10 10  : C5
0DA8  C8 65 02 49 38 A8 23 11  : 8C
0DB0  83 40 02 02 19 04 98 4C  : C8
0DB8  10 10 C8 64 42 29 0C 92  : 55
0DC0  23 11 80 04 04 12 21 10  : FF
0DC8  40 48 26 0B 0A 65 00 68  : 90
0DD0  80 98 44 10 11 05 84 80  : 86
0DD8  68 80 98 44 10 11 45 84  : AE
0DE0  C0 68 80 86 41 26 13 04  : AC
0DE8  04 32 19 1C 86 48 20 94  : ED
0DF0  84 62 30 68 80 92 4E 21  : FF
0DF8  88 09 04 C0 68 80 98 44  : 19
-----------------------------------
SUM:  E7 14 52 6B 4B 62 E3 4E  9EF7

0E00  10 10 45 84 C0 68 80 98  : 29
0E08  44 10 0A 08 84 51 48 B0  : 33
0E10  82 0D 00 08 08 64 12 61  : 76
0E18  30 40 43 21 92 49 C4 32  : A5
0E20  18 8C 46 00 10 10 48 84  : D6
0E28  41 01 20 98 2C 29 94 01  : E4
0E30  A2 02 61 10 40 44 16 12  : C1
0E38  01 A2 02 61 10 40 45 16  : B1
0E40  13 01 A2 02 19 04 98 4C  : B9
0E48  10 10 C8 64 72 19 20 82  : 79
0E50  52 11 88 C1 A2 02 21 14  : 85
0E58  86 20 24 13 01 A2 02 61  : E3
0E60  10 40 41 16 13 01 A2 02  : 5F
0E68  61 10 40 28 22 11 45 22  : 73
0E70  C2 08 34 00 20 21 90 49  : 18
0E78  84 C1 01 0C 86 44 22 90  : CE
-----------------------------------
SUM:  B4 F9 27 42 73 5B 49 C8  D4C3

0E80  C8 62 31 18 00 40 41 22  : 16
0E88  11 04 04 82 60 B0 88 45  : 78
0E90  06 88 0A 04 F2 80 80 88  : 16
0E98  45 06 88 08 64 12 61 30  : E2
0EA0  40 43 21 94 09 24 E2 A0  : E7
0EA8  8C 46 0D 00 08 08 64 12  : 65
0EB0  61 30 40 43 22 14 08 85  : D7
0EB8  02 48 8C 46 00 10 10 48  : 84
0EC0  84 41 01 20 98 2C 22 11  : DD
0EC8  41 A2 02 81 3C A0 20 22  : 84
0ED0  11 41 A2 02 61 10 40 41  : E8
0ED8  16 13 01 A2 02 61 10 40  : 7F
0EE0  28 22 11 45 22 C2 08 34  : C0
0EE8  00 20 21 90 49 84 C1 01  : 60
0EF0  0C 88 50 22 14 08 62 31  : B5
0EF8  18 00 40 50 27 94 04 04  : 6B
-----------------------------------
SUM:  8B F6 29 4F C6 F1 C9 BC  B5F7

0F00  42 28 34 40 43 20 93 09  : DD
0F08  82 02 19 0C A0 49 27 15  : CE
0F10  04 62 30 68 00 40 43 20  : A1
0F18  93 09 82 02 19 0C A0 44  : 29
0F20  28 12 44 62 30 04 B7 A8  : 73
0F28  2C FC A7 64 08 C3 0F A0  : AD
0F30  D2 9D 90 BE 60 04 C4 61  : 46
0F38  87 BB 36 A0 06 08 41 A0  : 07
0F40  07 2B 34 00 00 60 B2 5C  : D4
0F48  A2 1A 52 18 CD A8 01 8E  : 2A
0F50  17 28 44 14 00 5B BB 4A  : F7
0F58  4D CA 58 18 E1 72 84 4C  : AA
0F60  80 05 BB B4 A4 DC A5 81  : 9A
0F68  8E 17 2B 36 92 03 30 59  : 24
0F70  24 32 18 01 CA 21 18 00  : 72
0F78  1C AC DA 02 18 E1 70 B2  : BF
-----------------------------------
SUM:  63 2C AA 0B 60 3E B7 D7  72BE

0F80  59 B4 14 31 9B 79 C0 18  : 3E
0F88  76 C8 1C 45 58 83 04 20  : 9E
0F90  10 03 95 9A 00 00 30 45  : B7
0F98  55 88 30 FC A7 64 09 80  : 9D
0FA0  06 F7 2B 36 99 06 30 45  : 72
0FA8  55 88 30 46 22 AC 41 81  : E3
0FB0  87 4C B5 90 31 82 E1 72  : 1E
0FB8  B1 44 82 51 40 64 F8 FE  : 62
0FC0  50 39 42 23 00 03 55 A9  : EF
0FC8  55 83 1A 2E 13 E3 95 9B  : 46
0FD0  67 C3 1C 27 D6 6D 9F 0C  : 5B
0FD8  0C 04 4F AF E5 03 94 22  : AC
0FE0  30 00 35 5A 95 D4 31 A2  : FB
0FE8  E1 3E B3 6C F8 60 84 90  : AA
0FF0  00 72 B3 6C 70 63 05 92  : FB
0FF8  F5 07 83 C1 E0 F6 6D A8  : 2B
-----------------------------------
SUM:  E5 50 6C 83 71 DB 8B 11  B20E

1000  0C 78 B9 81 EC 61 83 F8  : 86
1008  74 38 01 31 80 E2 60 01  : A1
1010  BD CA CD B3 61 8E 16 4C  : 58
1018  64 10 03 21 22 A3 A6 20  : 23
1020  A1 47 7D 40 F2 1A 0A 62  : 1D
1028  62 9A A4 64 FC 13 21 A1  : D5
1030  A2 21 A0 A6 46 4E C1 32  : 90
1038  1A 1A 3A 4A 72 A4 64 A8  : DA
1040  13 21 A6 22 A0 A9 29 C6  : 34
1048  4E C3 1A 1A 7A 64 74 28  : BF
1050  19 A1 A7 A6 C6 41 01 6A  : 79
1058  1A 7A 72 9A A4 64 7C 22  : 46
1060  21 A7 AA AA 46 50 C0 22  : 94
1068  1A 82 A2 94 65 F0 11 A1  : D9
1070  A9 A8 47 53 41 9A 22 2A  : 12
1078  1C 75 0C 13 22 22 A3 29  : C0
-----------------------------------
SUM:  F4 EB FD 3A 27 41 9F D2  CDFF

1080  AA 27 C7 55 41 72 22 7A  : 3C
1088  AA 12 62 2C 77 64 02 A2  : C9
1090  A7 2A 29 2C C7 5D C1 9A  : A5
1098  2A 89 84 65 8C 0B 22 AC  : 01
10A0  28 26 22 AB 47 7B 00 32  : 0F
10A8  2A C2 A2 2A 92 74 75 98  : CB
10B0  16 A3 23 20 A2 22 47 70  : 77
10B8  C1 6A 32 32 0A 72 24 77  : A6
10C0  B8 16 A3 23 20 A9 A6 47  : 4A
10C8  78 01 6A 32 32 0A 9A 94  : 7F
10D0  77 60 0F 23 23 21 A0 A6  : 93
10D8  26 47 65 01 6A 32 32 22  : C3
10E0  4A B4 75 A8 1A 23 23 22  : 9D
10E8  A8 C7 65 C1 E2 32 32 3A  : 15
10F0  2C 75 24 1E 23 23 23 AA  : F6
10F8  47 40 01 E2 32 32 62 2C  : 5C
-----------------------------------
SUM:  80 CF 6F 1B C0 71 D3 E8  1614

1100  77 3C 1A 23 23 26 2A 47  : AA
1108  68 81 6A 32 32 6A 7A 24  : BF
1110  76 14 17 23 23 26 AA A6  : 5D
1118  2A 47 67 41 A2 32 32 72  : 91
1120  2C 76 A0 16 23 23 27 A9  : 6E
1128  47 7A 80 EA 32 32 92 2A  : 4B
1130  A4 75 DC 16 A3 23 29 AA  : A4
1138  A1 47 6D 41 6A 32 32 C2  : 26
1140  7A 94 65 30 03 23 24 A7  : 94
1148  22 23 A6 47 69 01 DA 3A  : B0
1150  29 84 64 04 03 23 A2 AA  : 87
1158  26 20 A1 47 72 00 72 3A  : 4C
1160  2A A2 62 7A 1C 74 60 1A  : B2
1168  A3 A2 AA 26 A2 47 5A 80  : D8
1170  72 3A 2A A2 6A 2A 6C 75  : ED
1178  58 1D A3 AA 18 47 40 00  : 61
-----------------------------------
SUM:  B9 BA 54 BE 9D 05 0C 96  C76D

1180  32 42 2A 0A 22 2A 94 76  : FE
1188  EC 0A A4 A6 A6 A2 A2 47  : 71
1190  72 40 B2 4A 6A 6A 2A 21  : CD
1198  94 74 C0 19 A4 A7 21 C7  : 14
11A0  7F 40 32 4A 72 22 4A 92  : AB
11A8  2C 60 00 02 A4 AA 2C A8  : B0
11B0  22 C7 41 01 22 52 AA 6A  : B3
11B8  84 74 34 1D A6 22 98 46  : EF
11C0  4D C0 6A 62 2A B2 2A 64  : 43
11C8  77 44 16 26 27 22 A3 C7  : AA
11D0  5E 00 32 62 7A 0A 22 0A  : A2
11D8  94 77 70 19 A6 2A 18 46  : C2
11E0  78 C0 32 69 C1 82 32 62  : AA
11E8  3C 66 14 02 26 A0 A1 A7  : C6
11F0  47 5B 41 32 6A 7A 22 9A  : B5
11F8  A2 5C 76 54 0A 26 A7 AB  : 4A
-----------------------------------
SUM:  C8 33 06 71 80 E7 DC 58  48F2

1200  22 C7 6A 81 9A 72 29 84  : 8D
1208  77 F0 15 A7 22 A3 C6 4F  : FD
1210  83 12 72 64 64 AC 23 27  : C5
1218  A8 2A 24 A6 A4 C6 6D C3  : 36
1220  32 7A AA A2 12 CA A4 64  : DC
1228  94 33 27 AA AA 22 22 A1  : 27
1230  C7 59 03 32 7A AA A2 42  : 5D
1238  2A C4 64 6C 23 27 AA AA  : 5C
1240  28 2A AA 47 40 02 32 82  : 39
1248  2A 2A 82 42 7C 75 7C 12  : 97
1250  A8 27 A4 A7 2A 46 52 03  : DF
1258  22 82 7A 62 64 76 78 0D  : DF
1260  A8 27 A8 47 4D C3 32 82  : 82
1268  7A 9A A2 62 0C 74 14 33  : DF
1270  28 29 24 A7 2A 26 47 75  : 28
1278  C0 A2 82 AA 9A 44 77 B4  : 97
-----------------------------------
SUM:  A1 46 87 A8 84 18 0D 30  27B1

1280  07 28 2A AA 26 A2 A6 C7  : 38
1288  52 40 B2 82 AA A2 9A A2  : 4E
1290  5C 66 54 03 29 A2 A0 A9  : 2D
1298  21 A4 47 7C 00 B2 9A 6A  : 3E
12A0  0A 92 A2 84 67 68 03 29  : BD
12A8  A9 A7 20 A6 A2 C6 48 40  : 06
12B0  F2 9A A2 0A 3A 2A 74 77  : 87
12B8  1C 0D 29 AB C7 6A C0 A2  : 90
12C0  9A BA 0A 84 76 AC 0A A9  : B7
12C8  AB A0 A8 19 47 6D 40 F2  : F2
12D0  9A BA 0A 82 9A A4 64 48  : CA
12D8  03 29 AC A6 AA 20 A1 47  : 30
12E0  6A 81 B2 A2 2A 9A A2 52  : F7
12E8  AC 74 1C 03 2A 29 20 A4  : 56
12F0  A6 22 C7 7C 81 DA AA 3A  : 4A
12F8  2C 77 90 1D AA A3 AA 47  : 8E
-----------------------------------
SUM:  61 1D 91 8D 83 77 5E 9F  2A66

1300  75 81 DA AA 62 2C 77 20  : 9F
1308  1D AA A6 2A 47 41 01 22  : 42
1310  AA 62 A1 84 74 64 13 2D  : 49
1318  22 A9 27 A5 2A CE 00 00  : 8F
1320  00 9E 1A A2 12 CA A4 64  : 3E
1328  94 33 27 AA AA 22 22 A1  : 27
1330  C7 59 03 32 7A AA A2 42  : 5D
1338  2A C4 64 6C 23 27 AA AA  : 5C
1340  28 2A AA 47 40 02 32 82  : 39
1348  2A 2A 82 42 7C 75 7C 12  : 97
1350  A8 27 A4 A7 2A 46 52 03  : DF
1358  22 82 7A 62 64 76 78 0D  : DF
1360  A8 27 A8 47 4D C3 32 82  : 82
1368  7A 9A A2 62 0C 74 14 33  : DF
1370  28 29 24 A7 2A 26 47 75  : 28
1378  C0 A2 82 AA 9A 44 77 B4  : 97
-----------------------------------
SUM:  09 AD 2A 73 07 30 19 E2  5BCE
```

リスト3 RDRTL.REL

```
0000  85 54 91 14 95 13 20 55  : 9B
0008  24 45 25 44 C8 15 55 31  : 35
0010  25 39 2E 06 5A 5A 53 54  : ED
0018  41 4B 81 56 96 93 51 53  : 30
0020  60 44 55 84 95 48 15 19  : 88
0028  3D 41 15 3A 06 46 43 4C  : A8
0030  4F 53 45 81 50 50 93 D4  : 6F
0038  95 20 64 75 24 14 24 94  : 7E
0040  F8 19 19 49 15 15 25 3E  : 00
0048  05 5F 52 45 41 44 81 10  : 11
0050  D4 15 55 20 65 F5 75 24  : 51
0058  95 44 58 15 19 1D 15 51  : E2
0060  0E 04 47 45 54 43 81 91  : 47
0068  D1 55 10 D2 10 60 44 74  : 30
0070  55 45 38 15 19 1D 15 51  : 83
0078  4E 05 46 50 55 54 43 81  : 56
-----------------------------------
SUM:  78 89 65 A7 02 86 75 94  6791

0080  14 15 55 10 E0 65 05 55  : 2D
0088  44 34 84 18 11 41 55 51  : 0C
0090  4E 05 46 50 55 54 53 81  : 66
0098  97 D5 13 54 14 91 60 65  : 3D
00A0  F5 35 15 75 24 99 40 00  : B1
```

▶「遥かなるオーガスタ」皆さん買いましたか？　私は買ったその日に－4を出して以来スコアが縮まりません。
湯川　光雄(22)茨城県

```
00A8  04 D4 BC 21 DA 73 A5 A1  : 48
00B0  00 60 40 86 48 7C 02 00  : EC
00B8  00 04 47 E0 11 F2 00 EE  : 1C
00C0  19 08 3F 04 62 2A 56 10  : 56
00C8  3A 2E 87 C6 54 5C 21 DA  : 60
00D0  7B 35 07 C4 20 10 00 8A  : 35
00D8  94 84 03 80 02 1A D0 00  : 87
00E0  ED 2D 9D 83 E1 30 98 68  : 4B
00E8  26 A7 14 06 DF C2 01 43  : CC
00F0  DD FC 3E 65 59 80 06 6D  : C8
00F8  60 00 0D 04 D4 E2 80 53  : FA
--------------------------------------
SUM:  E8 4F 56 C8 76 09 5A FA  4A35

0100  F8 40 20 7B 86 D5 E1 20  : 2F
0108  98 61 C0 06 00 31 44 35  : 69
0110  A0 01 CA 3E 01 33 40 00  : 1D
0118  06 1D 37 01 0D 9C C4 67  : 2F
0120  21 18 4C 18 C9 00 00 00  : 66
0128  00 00 00 00 00 00 00 00  : 00
0130  00 00 00 00 00 00 00 00  : 00
0138  00 00 00 00 00 00 00 00  : 00
0140  00 00 00 00 00 00 00 00  : 00
0148  00 00 00 00 00 00 00 00  : 00
0150  00 00 00 00 00 00 00 00  : 00
0158  00 00 00 0D 3F 87 CC 2B  : CA
0160  48 00 3E 07 8C A9 30 40  : 32
0168  4D 8A E5 66 DD 60 06 6D  : D2
0170  E1 00 6A 88 68 20 0B 94  : FA
0178  75 49 82 1F C0 F1 0D 00  : 1D
--------------------------------------
SUM:  42 AA 3C F9 2D 76 43 28  CCCC

0180  01 10 01 D5 E3 2A 4C 10  : 50
0188  21 A0 40 2E 56 6D 98 01  : 8B
0190  60 B0 4F 8B 56 55 F3 00  : 88
0198  11 52 90 86 8B 85 82 C3  : CE
01A0  A6 20 01 A5 3B 2A F9 80  : 4A
01A8  3F 84 0C 00 98 61 EA 21  : D3
01B0  A0 80 20 60 80 68 6C 00  : F4
01B8  7F 08 19 0A 76 41 DC 26  : 63
01C0  23 08 3C C2 34 50 05 9A  : 4C
01C8  E5 0F B0 E9 B8 08 00 00  : 4D
01D0  52 00 15 C0 04 10 00 00  : 3B
01D8  00 00 00 00 00 00 00 00  : 00
01E0  00 00 00 00 00 00 00 00  : 00
01E8  01 48 88 52 2A 13 07 40  : A7
01F0  92 A9 38 82 42 26 11 44  : B2
01F8  05 42 78 80 9E 50 22 93  : E2
--------------------------------------
SUM:  89 28 9F E2 DD 96 C3 4C  256B

0200  84 04 F2 A9 50 A0 55 2A  : 92
0208  08 08 C4 92 61 14 1A 00  : F5
0210  15 52 90 83 E2 DC 50 05  : 8D
0218  72 B3 6B 10 13 04 75 45  : 71
0220  C2 06 45 D0 FB B4 F7 4B  : CE
0228  42 18 6F A0 FB 05 C2 7C  : A7
0230  5A 8A 00 87 D1 95 6C 01  : 3E
0238  08 D6 30 16 6B 94 3E C3  : 24
0240  A6 E0 20 D2 09 08 9E 52  : 79
0248  2A 11 48 82 00 38 34 00  : 71
0250  06 04 08 64 87 C0 34 1C  : 0D
0258  00 22 3F 00 8F 90 0B E1  : 6C
0260  40 1C 32 10 7D 08 40 00  : 63
0268  06 48 D8 00 11 61 3F C3  : 9A
0270  2C 96 0B 85 CA C5 08 8F  : 78
0278  80 01 91 B2 00 42 00 00  : 06
--------------------------------------
SUM:  41 A1 EA DA 4F 76 2F A0  85FD

0280  32 58 2E 16 88 8A 8F 04  : 73
0288  3B 4A 74 70 21 CA D5 62  : 8B
0290  B3 6B 78 08 8A 8C 04 1F  : D7
0298  16 AC 87 2B 75 C2 DD 1B  : A3
02A0  0F 40 0D D1 B0 F0 00 11  : DE
02A8  1F 80 03 2D D3 A8 E5 BA  : E9
02B0  74 1D 0A A8 F0 41 F9 CC  : 39
02B8  5F 7F 14 98 57 8C 04 7C  : ED
02C0  04 76 96 E8 E0 40 AA 8C  : 4E
02C8  04 33 69 C0 20 E0 FC DD  : 39
02D0  1B 0E C0 0C 3A AC 04 E6  : C5
02D8  2F BF 8A EC AA 06 04 FE  : 16
02E0  20 B0 AA D0 10 F8 08 ED  : 47
02E8  2D D1 C0 81 55 18 08 66  : 1A
02F0  D3 80 46 95 11 02 15 51  : A7
02F8  E0 81 B1 5C 7C 04 76 96  : FA
--------------------------------------
SUM:  89 0D 79 D9 48 EF 70 3A  F3B5

0300  E8 E0 40 AA 8C 04 33 69  : DE
0308  08 28 E0 92 DD 1B 0E C0  : 68
0310  0D D3 F0 F9 EC 01 6E 9D  : C1
0318  C7 C1 11 F8 00 55 46 02  : 2E
0320  03 2D D3 A8 E5 BA 74 1D  : DB
0328  0A A8 F0 41 F9 CC 5F 7F  : 86
0330  10 44 02 26 E9 F8 6C DD  : A6
0338  3B 8C 99 B7 F8 14 70 1B  : AE
0340  6E 8D 06 42 AA 30 10 11  : 3E
0348  1F 80 03 2E D2 DD F1 07  : 77
0350  0B 00 03 2D D3 A8 E5 BA  : 55
0358  74 1D 0A A8 C0 43 25 BA  : 25
0360  36 1F 20 04 20 00 03 25  : C1
0368  82 E1 11 51 80 87 2B 14  : 0B
0370  F8 B7 10 80 00 0C 86 E8  : B9
0378  AA 8C 04 37 4F C3 E7 30  : 9A
--------------------------------------
SUM:  82 AE DA 44 12 55 4A 39  051E

0380  04 50 2A 10 80 00 0E 56  : 72
0388  EB 95 9B 68 C0 DB AE 16  : E2
0390  0B 74 FC 3B 6E 9D C2 4A  : CD
0398  71 40 31 DA 5B A3 01 0C  : C7
03A0  DA CA 0C 38 0A 37 46 83  : F2
03A8  27 69 6E 8C 04 33 6C D0  : FD
03B0  28 E0 10 DD 1B 0F 90 02  : B1
03B8  10 08 01 92 DD 1B 0F 90  : 42
03C0  02 10 00 01 92 C1 70 B9  : 8F
03C8  58 A7 C5 BB 2A 89 81 B9  : 6C
03D0  5B AE 16 E9 F8 7C E6 01  : 63
03D8  08 5F EF F6 03 74 FC 3D  : FC
03E0  53 8A 01 0D D1 B0 F4 00  : 60
03E8  6F 13 00 19 2D D3 70 E5  : F0
03F0  BA 66 1D 37 4F C3 B5 38  : 73
03F8  80 28 DD 72 B4 59 B7 F8  : B3
--------------------------------------
SUM:  5D A3 42 2A C7 88 73 6C  C74C

0400  14 42 FF 7F B6 1B A3 41  : 89
0408  93 75 CA E1 08 8F C0 01  : 0B
0410  96 E8 D4 76 7E 11 B7 4E  : 5C
0418  A3 96 E9 D0 74 A7 37 89  : CD
0420  80 0C 06 E9 F8 7C F6 01  : E6
0428  37 4E E3 E1 0B FD FE C9  : 18
0430  66 88 44 0C D7 A0 7C DE  : 0F
0438  26 00 3F 83 4C 01 0B FD  : 3D
0440  FE C9 60 B4 5A AC 56 6B  : A2
0448  4C 3E 1A 7F 06 D9 54 DC  : 32
0450  05 9A EE 0F 98 8D 6C 96  : C3
0458  8B 75 C2 C1 70 B9 58 AD  : B1
0460  D7 2B 54 45 49 C2 01 67  : 0E
0468  85 8B 2A B8 81 B1 5C AD  : 2D
0470  D7 2B 36 DA 01 1F 16 E7  : 2F
0478  D6 EB 85 C2 C1 65 56 A0  : 24
--------------------------------------
SUM:  06 F9 55 9B CA 3E 03 E3  2B2A

0480  37 69 6E 93 84 39 56 EE  : A2
0488  D2 93 85 85 5C 40 C4 20  : EF
0490  00 03 24 EE 23 7F 03 58  : 12
0498  55 0C 0C 6C 00 15 52 70  : B0
04A0  86 4B 05 A2 E1 72 B5 58  : D8
04A8  A7 A5 B8 80 28 7B 7F 00  : A6
04B0  44 00 67 2B 36 F4 01 B8  : B9
04B8  59 2F E0 11 F5 94 F4 0F  : 05
04C0  B0 EC F8 1B 95 AA CD B4  : 6F
04C8  60 6D 13 E2 DD C2 C8 10  : 39
04D0  BF DF EC 96 0B 75 C2 E1  : 43
04D8  72 B7 5C AC 53 E8 CA 90  : C6
04E0  04 37 44 54 60 21 BA 6E  : 7C
04E8  1C B7 4C C3 A3 B8 8D BA  : 84
04F0  75 1C B7 4E 83 A6 E8 D0  : 77
04F8  76 20 0F 3B 4B 74 60 21  : 20
--------------------------------------
SUM:  74 43 D0 AF D8 3E 48 43  9EDB

0500  9B 59 41 84 2F F7 FB 61  : 3B
0508  BA 34 19 37 5C AE 10 88  : E0
0510  FC 00 19 6E 9D 47 2D D3  : 67
0518  A0 E8 75 48 02 04 C0 03  : 0E
0520  7B 25 82 E1 72 B1 47 54  : C1
0528  A8 21 4E 28 06 39 5D A5  : 80
0530  BA 52 10 D5 66 D7 70 36  : D4
0538  0B 05 92 7D 66 BD 03 E2  : 27
0540  60 03 25 82 E1 72 B1 5C  : 6A
0548  A2 AA 52 10 E5 66 D1 40  : 0A
0550  46 0B 05 92 C1 68 B8 5C  : 25
0558  AD 56 29 F8 46 B7 65 53  : D9
0560  40 47 29 3C 0C 00 62 B5  : 0F
0568  59 B5 DC 0C F8 B7 68 B0  : BD
0570  5C 22 87 50 85 FE FF 64  : 3B
0578  88 40 00 06 48 8A EE 03  : 91
--------------------------------------
SUM:  4B 7E 8B 86 0C A4 65 E7  3AC6

0580  B3 54 61 F6 C3 36 D0 02  : 29
0588  36 05 47 40 F8 46 EF 83  : 72
0590  80 44 00 07 6A C0 74 5D  : C6
0598  0F 8C A8 80 42 BC 65 77  : 9D
05A0  81 D9 B5 DC 1D B0 06 08  : C6
05A8  03 80 03 AB FA 0E 11 BF  : 09
05B0  E0 E1 20 9B F8 FE 30 07  : A9
05B8  8A 8C 41 F4 29 BC 60 01  : 91
05C0  12 1F 80 A2 81 20 30 30  : 54
05C8  EB 06 BD 4F 28 74 3A 1D  : F0
05D0  0E 4F 78 B5 90 08 11 95  : C8
05D8  16 08 57 8C A8 88 40 04  : 75
05E0  70 00 76 96 EE E0 38 86  : 08
05E8  EF 83 BB 56 0B 76 48 F8  : 44
05F0  0E 37 64 8E 8B A1 F6 6D  : C6
05F8  BF 04 6C 33 6E 50 23 60  : A3
--------------------------------------
SUM:  B3 29 76 B2 72 DB 93 59  A266

0600  7C 08 1B B0 1C AE D2 D9  : C4
0608  D0 3E 01 10 00 1D AB 07  : EE
0610  09 F9 9B 45 01 59 2F E2  : 4D
0618  08 E0 08 FE 22 8F DA 03  : 7C
0620  E0 1B 24 1C 10 76 96 CC  : 23
0628  01 F7 69 4E 89 04 0A 8C  : D2
0630  81 F1 F0 05 9A 00 10 36  : 47
0638  00 C0 81 15 14 08 3F 3F  : F0
0640  9F E2 80 D2 DC 50 0B 6A  : 74
0648  BB 4B 67 40 FB 36 82 82  : E2
0650  B4 45 00 D6 A8 44 40 00  : FB
0658  0C B4 42 0E 20 98 34 40  : 3C
0660  CF 1F 2B D6 EC 96 2B 94  : 30
0668  0C 10 09 88 C3 4B E1 00  : 9C
0670  08 20 F8 70 B0 59 2E 57  : 1E
0678  32 1C 42 1F 14 AF 9C 2C  : 3A
--------------------------------------
SUM:  EE 73 54 6A 98 80 4C D5  0ED1

0680  81 F0 18 6E C9 11 5D C0  : EE
0688  76 6A 8C 3E D8 66 AB C3  : 56
0690  ED 80 08 80 00 11 BB E0  : A1
0698  E2 A3 A0 7D DA B0 1D 17  : 60
06A0  43 E3 2A 20 10 2A 70 89  : A3
06A8  C4 54 50 20 54 DF 13 88  : 56
06B0  A8 90 40 84 00 00 11 5F  : 6C
06B8  10 76 6D A6 07 6C 0C AF  : C7
06C0  E8 38 CA FF 83 8F 90 03  : 8E
06C8  2A 00 10 AF 19 51 60 81  : 34
06D0  95 DE 07 19 51 10 81 08  : 7D
06D8  00 00 22 BE 20 E0 11 00  : F1
06E0  01 DA 5B A2 81 02 A3 A0  : 9E
06E8  7D DA B0 15 51 40 81 B0  : DE
06F0  14 7C 01 76 96 E8 90 40  : 55
06F8  A8 C8 1F 66 80 C4 0D 80  : C6
--------------------------------------
SUM:  66 C8 A1 2B DB 6B C3 35  2D7B

0700  08 E0 00 ED 2D DD C0 71  : 10
0708  0D DF 07 76 AC 16 EC 97  : AE
0710  58 46 EF 83 80 47 00 07  : DE
0718  6A C0 75 44 02 06 45 D0  : 00
0720  F8 EA 8B 04 0A AF 88 38  : EA
0728  B0 5A 20 00 8F 0C E2 2B  : D2
0730  E2 0E 3E 00 BB 4B 74 48  : F0
0738  20 54 64 0F B3 40 02 06  : E2
0740  C0 86 EF 83 BB 4B 74 50  : 82
0748  20 02 20 00 3B 56 03 E0  : B6
0750  0B B4 B7 44 82 05 46 40  : C7
0758  FB 34 06 20 6C 01 82 02  : 46
0760  1B FE 0E 7E 7F 1F C6 01  : 0A
0768  21 19 39 CA 2A 31 07 C2  : 61
0770  C0 02 F8 64 E2 E1 02 B2  : 95
0778  AC E0 20 61 D2 CD B2 A0  : FE
--------------------------------------
SUM:  0F D4 E3 31 A3 2B 91 17  694B

0780  EC 90 08 E0 00 EB 08 DD  : 34
0788  F0 77 6A C0 75 44 02 06  : 52
0790  45 D0 F8 EA 88 84 11 C7  : DB
0798  54 58 21 70 DA B3 80 87  : D1
07A0  86 6D 5B 07 75 B5 5B AE  : 88
07A8  57 08 44 7E 00 0C B4 47  : 28
07B0  C0 16 68 00 40 C9 00 8E  : D5
07B8  00 02 2B B8 0E 11 BB E0  : 9F
07C0  EE D5 80 EA 88 04 0C 8B  : 50
07C8  A1 F1 D5 11 08 23 8E A8  : D9
07D0  B0 42 E0 70 1A 3C 33 6A  : 35
07D8  D8 3B AC 7C 01 10 D7 20  : 43
07E0  16 68 00 40 38 05 8A AE  : 33
07E8  E0 38 44 62 00 1A 08 40  : 20
07F0  00 06 48 85 FE FF 64 80  : B4
07F8  47 00 07 58 8A EE 03 84  : A5
--------------------------------------
SUM:  66 A5 31 9D 05 80 02 43  DAB8

0800  6E F8 3B B5 60 3A A2 01  : 93
0808  03 22 E8 7D 9B 77 81 9B  : B8
0810  03 AA 22 10 CD AB 60 EE  : A5
0818  B6 AB 75 CA E1 08 8F C0  : D8
0820  01 96 88 F8 02 CD 01 88  : 6F
0828  1B 00 11 C0 01 DA 5B BB  : DD
0830  80 E2 1B BE 0E ED 58 2D  : BB
0838  D9 20 11 C0 00 45 77 01  : 87
0840  C2 37 7C 1D DA B0 1D 51  : 8A
0848  00 81 91 74 3E CD BB C0  : 0C
0850  C3 81 28 EA 88 84 33 6A  : FF
0858  D8 3B AC 7C 01 10 D4 F0  : 19
0860  66 68 0C 40 38 0A 08 68  : CC
0868  88 40 D0 02 38 00 3B 4B  : 58
0870  77 70 1C 43 77 C1 DD AB  : 06
0878  01 08 00 00 C9 10 BF DF  : 80
--------------------------------------
SUM:  62 9B 58 BE 0B 29 FB 63  8314

0880  EC 91 D5 11 08 23 8E A8  : C4
0888  B0 42 E1 A0 3A A2 21 0E  : 7E
0890  67 81 1C 75 45 82 1C CF  : 2B
0898  05 C0 C0 90 CB 1F B2 C7  : 78
08A0  EC B1 FB 2C 7E 21 A0 01  : 04
08A8  01 60 01 7C 32 CD AE E0  : 6B
08B0  ED 87 28 EA 8B 04 39 9E  : EC
08B8  0C DA B6 0E CD B9 00 E2  : 12
08C0  A3 10 7C 2C 00 2F 86 46  : 56
08C8  C8 F6 6D A6 07 3B B8 5B  : 26
08D0  07 71 51 88 3E 16 00 17  : BC
08D8  C3 23 64 00 EA 8B 04 39  : FC
08E0  9E 0C 60 80 CA 8B 04 33  : 16
08E8  6C A8 3B 60 30 A1 1D 51  : EE
08F0  10 81 95 16 08 65 8F D9  : 11
08F8  63 F6 58 FD 96 3F 10 D0  : 63
--------------------------------------
SUM:  A0 4B 92 A3 21 EC 06 CB  EB7C

0900  00 80 B0 00 BE 19 1D 51  : 75
0908  10 87 30 3D 8C 80 3B B2  : FD
0910  5E AF 56 58 FD 96 3F 65  : F2
0918  8F D9 63 F1 0D FF 07 0B  : DA
0920  00 0B E1 93 F3 36 E0 03  : 8B
0928  BC 5C C0 F4 29 BD E2 C9  : 5D
0930  6A B9 47 57 78 1C AE 3A  : 3D
0938  2E 87 D7 42 80 78 CA EF  : 7F
0940  03 8F 80 2E D2 D9 78 3E  : A1
0948  2A 31 07 D9 A0 01 03 85  : 64
0950  A2 C9 6A B9 47 C0 17 69  : 15
0958  6C BC 1F 15 18 83 EC D0  : B3
0960  18 81 C2 D1 64 B1 5C A0  : 3D
0968  64 00 A8 C4 1F 3F 2D C5  : 20
0970  00 81 18 41 F2 3E 04 8D  : 9B
0978  C3 00 41 F2 01 20 B7 70  : 3E
--------------------------------------
SUM:  CB 7D 2B 43 AF 20 9A C6  EDD7

0980  B0 59 22 60 01 BC 52 29  : C3
0988  14 8A 59 2E 56 58 F1 96  : 5A
0990  1D 65 8F 19 61 D6 58 F1  : AA
0998  96 1D 65 8F 19 61 D3 EB  : DF
09A0  85 92 00 00 00 00 00 00  : 17
09A8  00 00 00 00 00 00 00 00  : 00
09B0  00 00 00 00 00 00 00 00  : 00
09B8  00 00 00 00 00 00 00 00  : 00
09C0  00 00 00 00 00 00 00 00  : 00
09C8  00 00 00 00 00 00 00 00  : 00
09D0  00 00 00 00 00 00 00 00  : 00
09D8  00 00 00 00 00 00 00 00  : 00
09E0  00 00 00 00 00 00 00 00  : 00
09E8  00 00 00 00 00 00 00 00  : 00
```

▶無事，府立高校に入学できました。私の学校は明智光秀と豊臣秀吉の山崎の合戦で有名な天王山の「おなか」の場所にあります。竹林に囲まれて空気もいいし，高台からの景色は絶品です。　天達　雄一(15)京都府

```
09F0  00 00 00 00 00 00 00 00  : 00
09F8  00 00 00 00 00 00 00 23  : 23
--------------------------------------
SUM:  FC F7 6F 36 D1 4B 6E BE  63BA

0A00  BF C0 B5 7D 49 15 05 12  : 26
0A08  3B 3E 0D 97 D4 D4 55 D4  : EE
0A10  92 63 A5 80 D9 7D 51 35  : F6
0A18  41 49 16 3A CA 0D 97 D5  : 1D
0A20  D4 92 55 11 63 AA 00 35  : 0E
0A28  05 09 3D 49 52 3B 46 07  : 6E
0A30  10 D4 15 55 23 A6 E0 31  : 28
0A38  15 61 25 52 3A C4 05 91  : 81
0A40  90 D3 13 D4 D1 63 B6 80  : B4
0A48  55 19 1D 15 51 0E 3A 6C  : A5
0A50  07 51 91 D1 55 14 E3 B3  : B9
0A58  00 35 19 3D 41 15 3A 3A  : 55
0A60  EE 07 51 94 15 55 10 E3  : 37
0A68  A2 80 95 19 41 55 51 4E  : 05
0A70  3B 14 03 91 94 91 51 52  : AB
0A78  53 E3 B6 80 51 1D 15 51  : 40
--------------------------------------
SUM:  D5 6A C2 84 C5 B4 41 9B  3403

0A80  0E 3A 26 07 91 D1 55 10  : 3C
0A88  D2 10 63 A4 60 71 1D 15  : EC
0A90  51 4E 3A DE 03 91 D4 90  : AF
0A98  50 92 53 E3 2B 60 11 35  : E9
0AA0  05 25 3A 3A EE 07 14 15  : BC
0AA8  55 10 E3 BD 00 79 41 55  : 14
0AB0  51 0D 21 06 3A 0A 09 14  : E6
0AB8  15 55 14 E3 A0 00 15 49  : 5F
0AC0  11 49 51 32 3A 0A 01 55  : 6D
0AC8  53 12 53 92 E3 37 E0 D5  : 19
0AD0  69 69 09 55 1A 3A 56 11  : EB
0AD8  56 96 93 51 53 63 A5 A1  : CC
0AE0  19 69 69 4D 51 05 2E 72  : 2E
0AE8  00 00 9E 1A 00 00 00 00  : B8
0AF0  00 00 00 00 00 00 00 00  : 00
```

```
0AF8  00 00 00 00 00 00 00 23  : 23
--------------------------------------
SUM:  7D 84 AF 1D C2 96 D4 22  5B1E
```

リスト4 POLL.REL

```
0000  ED 7B 06 00 01 24 00 11  : A4
0008  D3 01 21 5C 00 ED B0 0E  : FC
0010  0F 11 5C 00 CD 05 00 FE  : 4C
0018  FF 28 7C 0E 14 11 5C 00  : 32
0020  CD 05 00 A7 20 71 0E 14  : 2C
0028  11 5C 00 CD 05 00 A7 20  : 06
0030  16 21 D0 01 34 21 80 00  : DD
0038  ED 5B D1 01 01 80 00 ED  : 88
0040  B0 ED 53 D1 01 18 DF 0E  : C7
0048  13 11 D3 01 CD 05 00 FE  : C8
0050  FF 28 4F 0E 16 11 D3 01  : 7F
0058  CD 05 00 FE FF 28 43 AF  : E9
0060  32 F3 01 21 F7 01 22 D1  : 32
0068  01 2A D1 01 11 80 00 01  : 8F
0070  80 00 ED B0 22 D1 01 0E  : 1F
0078  15 11 D3 01 CD 05 00 A7  : 73
--------------------------------------
SUM:  06 EB A7 91 16 E6 59 81  C773

0080  20 20 21 D0 01 35 20 E1  : 68
0088  0E 10 11 D3 01 CD 05 00  : D5
0090  FE FF 28 0E C3 00 00 0E  : 04
0098  09 11 AD 01 CD 05 00 C3  : 5D
00A0  00 00 0E 09 11 BB 01 CD  : B1
00A8  05 00 C3 00 00 4F 70 65  : EC
00B0  6E 20 45 72 72 6F 72 20  : B8
00B8  21 21 24 44 69 73 6B 20  : 11
```

```
00C0  41 63 63 65 73 73 20 45  : B7
00C8  72 72 6F 72 20 21 21 24  : 4B
00D0  00 F7 01 00 00 00 00 00  : F8
00D8  00 00 00 00 00 00 00 00  : 00
00E0  00 00 00 00 00 00 00 00  : 00
00E8  00 00 00 00 00 00 00 00  : 00
00F0  00 00 00 00 00 00 00 0E  : 0E
00F8  15 11 D3 01 CD 05 00 A7  : 73
--------------------------------------
SUM:  91 5E E7 49 DE 8C B4 42  6C37
```

リスト5 CUT.COM

```
0000  85 14 13 D3 13 20 45 04  : FB
0008  F4 C4 C9 40 00 04 D4 E4  : 7D
0010  01 82 E1 72 B1 4F 8B 56  : B7
0018  55 32 00 66 B4 03 EF E0  : 73
0020  98 80 24 CD 68 07 D4 E2  : 2E
0028  07 D0 84 00 00 7F 00 D8  : B2
0030  0E 56 68 00 00 C1 64 BF  : B0
0038  80 62 00 40 84 00 00 72  : 18
0040  B3 68 F0 03 04 4C 00 37  : 95
0048  B2 59 AD 00 F8 98 00 6F  : B7
0050  64 C6 4A C0 2A 0A 12 7A  : F4
0058  92 A4 74 00 02 28 27 A6  : A1
0060  26 4E 00 00 00 9E 1A 00  : 2C
0068  00 00 00 00 00 00 00 00  : 00
0070  00 00 00 00 00 00 00 00  : 00
0078  00 00 00 00 00 00 00 00  : 00
--------------------------------------
SUM:  7D 0D 28 BB 8C 71 1E CF  719A
```

リスト6 SC4.MAC

```
                        ; SMALL-C Compiler Version 2.7 (fas)
                        ; Copyright 1982, 1983 J. E. Hendrix
                        ; Part 4
                        ; Hand Compiled by T.Ishigami 1991
                        ;
                          EXT     macn
                          EXT     cptr
                          EXT     symtab
                          EXT     optimize
                          EXT     level
                          EXT     itype
                          EXT     explevel
                          EXT     stagenex
                          EXT     ssname
                          EXT     beglab
                          EXT     csp
                          EXT     output
                          EXT     m80flg
000D                    CR        EQU     0DH
0000'                   header::
0000'   CD 0000*          CALL    getlabel
0003'   22 0000*          LD      (beglab),HL
0006'   C9                RET
0007'                   trailer::
0007'   2A 0000*          LD      HL,(symtab)
000A'   11 0177           LD      DE,375
000D'   19                ADD     HL,DE
000E'   22 0000*          LD      (cptr),HL ;cptr = STARTGLB
0011'   2A 0000*        CC4:      LD      HL,(symtab)
0014'   11 0EAE           LD      DE,375 + 3383
0017'   19                ADD     HL,DE     ;hl = ENDGLB
0018'   ED 5B 0000*       LD      DE,(cptr)
001C'   A7                AND     A
001D'   ED 52             SBC     HL,DE
001F'   DA 0047'          JP      C,CC5     ;while (cptr<ENDGLB)
0022'   2A 0000*          LD      HL,(cptr) ;  if(cptr[IDENT]==FUNCTION)
0025'   7E                LD      A,(HL)
0026'   FE 04             CP      4
0028'   20 11             JR      NZ,CC6
002A'   23                INC     HL        ;  if(cptr[CLASS]==AUTOEXT)
002B'   23                INC     HL
002C'   7E                LD      A,(HL)
002D'   FE 04             CP      4
002F'   20 0A             JR      NZ,CC6
0031'   2A 0000*          LD      HL,(cptr) ;    external(cptr+NAME)
0034'   11 0008           LD      DE,8
0037'   19                ADD     HL,DE
0038'   CD 00EC'          CALL    external
003B'   2A 0000*        CC6:      LD      HL,(cptr)  ;cptr += SYMMAX
003E'   11 0011           LD      DE,17
0041'   19                ADD     HL,DE
0042'   22 0000*          LD      (cptr),HL
0045'   18 CA             JR      CC4
0047'   21 0068'        CC5:      LD      HL,CC3  ;ptr=findglb("main")
004A'   E5                PUSH    HL
004B'   CD 0000*          CALL    findglb
004E'   C1                POP     BC
004F'   7C                LD      A,H
0050'   B5                OR      L
0051'   28 0F             JR      Z,CC9
0053'   11 0006           LD      DE,6    ;ptr[OFFSET]==FUNCTION
0056'   19                ADD     HL,DE
0057'   7E                LD      A,(HL)
0058'   FE 04             CP      4
005A'   20 06             JR      NZ,CC9
005C'   21 006D'        CC11:     LD      HL,CC3+05H
005F'   CD 00EC'          CALL    external
0062'   21 0073'        CC9:      LD      HL,CC3+0BH
0065'   C3 0BF9'          JP      ol
0068'   6D 61 69 6E CC3:          DB      'main',00H
006C'   00
006D'   52 44 52 54       DB      'RDRTL',00H
0071'   4C 00
0073'   20 45 4E 44       DB      ' END',00H
0077'   00
0078'                   loadargc::
0078'   21 00C2'          LD      HL,CC12
007B'   E5                PUSH    HL
007C'   2A 0000*          LD      HL,(macn)
007F'   E5                PUSH    HL
0080'   21 000B           LD      HL,9 + 2
0083'   E5                PUSH    HL
0084'   2A 0000*          LD      HL,(macn)
0087'   11 0596           LD      DE,1430
008A'   19                ADD     HL,DE
008B'   E5                PUSH    HL
008C'   21 0082           LD      HL,130
008F'   E5                PUSH    HL
0090'   21 0000           LD      HL,0
0093'   E5                PUSH    HL
0094'   CD 0000*          CALL    search
0097'   EB                EX      DE,HL
0098'   21 000C           LD      HL,12
009B'   39                ADD     HL,SP
009C'   F9                LD      SP,HL
009D'   EB                EX      DE,HL
009E'   7C                LD      A,H
009F'   B5                OR      L
00A0'   C0                RET     NZ
00A1'   C1                POP     BC
00A2'   E1                POP     HL
00A3'   E5                PUSH    HL
00A4'   C5                PUSH    BC
00A5'   7C                LD      A,H
00A6'   B5                OR      L
00A7'   CA 00BC'          JP      Z,CC14
00AA'   21 00CB'          LD      HL,CC12+09H
00AD'   CD 0C40'          CALL    ot
00B0'   C1                POP     BC
00B1'   E1                POP     HL
00B2'   E5                PUSH    HL
00B3'   C5                PUSH    BC
00B4'   E5                PUSH    HL
00B5'   CD 0C64'          CALL    outhex
00B8'   C1                POP     BC
00B9'   C3 0000*          JP      nl
00BC'                   CC14:
00BC'   21 00D2'          LD      HL,CC12+10H
00BF'   C3 0BF9'          JP      ol
00C2'   4E 4F 43 43 CC12:         DB      'NOCCARGC',0
00C6'   41 52 47 43
00CA'   00
00CB'   20 4C 44 20       DB      ' LD A,',0
00CF'   41 2C 00
00D2'   20 58 4F 52       DB      ' XOR A',0
00D6'   20 41 00
00D9'                   entry::
00D9'   21 0000*          LD      HL,ssname
00DC'   CD 0C40'          CALL    ot
00DF'   CD 0000*          CALL    col
00E2'   3A 0000*          LD      A,(m80flg)
00E5'   A7                AND     A
00E6'   C4 0000*          CALL    NZ,col
00E9'   C3 0000*          JP      nl
00EC'                   external::
00EC'   E5                PUSH    HL
00ED'   21 00F7'          LD      HL,CC17
00F0'   CD 0C40'          CALL    ot
00F3'   E1                POP     HL
00F4'   C3 0BF9'          JP      ol
00F7'   20 45 58 54 CC17:         DB      ' EXT ',0
00FB'   20 00
00FD'                   indirect::
00FD'   C1                POP     BC
00FE'   E1                POP     HL
00FF'   E5                PUSH    HL
0100'   C5                PUSH    BC
0101'   22 0168'          LD      (lval),HL
0104'   5E                LD      E,(HL)
0105'   23                INC     HL
0106'   56                LD      D,(HL)  ;later DE means sym
0107'   21 0003           LD      HL,3
010A'   19                ADD     HL,DE   ;indlevel = sym[LEVEL]
010B'   4E                LD      C,(HL)
010C'   0C                INC     C
010D'   0D                DEC     C
010E'   CA 0141'          JP      Z,CC13
0111'   21 0002           LD      HL,2
0114'   19                ADD     HL,DE
0115'   7E                LD      A,(HL)
0116'   FE 02             CP      2
0118'   20 01             JR      NZ,inl  ;if(sym[CLASS] == AUTOMATIC)
011A'   0C                INC     C       ;       indlevel++;
011B'   21 0000*        inl:      LD      HL,level
011E'   7E                LD      A,(HL)
011F'   34                INC     (HL)
0120'   B9                CP      C
0121'   38 13             JR      C,CC19  ;wether level++ >= indlevel
0123'   1A                LD      A,(DE)
0124'   FE 03             CP      3
0126'   20 0E             JR      NZ,CC19 ;wether sym[IDENT] == POINTER
0128'   21 0004           LD      HL,4
012B'   19                ADD     HL,DE
012C'   7E                LD      A,(HL)          ;A = sym[ITYPE]
012D'   2A 0168'          LD      HL,(lval)
0130'   23                INC     HL
0131'   23                INC     HL
0132'   77                LD      (HL),A
0133'   23                INC     HL
0134'   36 00             LD      (HL),0
0136'   3A 0000*        CC19:     LD      A,(level)
0139'   B9                CP      C
013A'   38 05             JR      C,CC13
013C'   3E 01             LD      A,1
013E'   32 0000*          LD      (explevel),A
0141'   2A 0168'        CC13:     LD      HL,(lval)
0144'   23                INC     HL
0145'   23                INC     HL
0146'   7E                LD      A,(HL)
0147'   FE 04             CP      4
0149'   21 0155'          LD      HL,CC18
014C'   CA 03D8'          JP      Z,ffcall
014F'   21 015F'          LD      HL,CC18+10
0152'   C3 03D8'          JP      ffcall
0155'   43 43 47 43 CC18:         DB      'CCGCHAR##',0
0159'   48 41 52 23
015D'   23 00
015F'   43 43 47 49       DB      'CCGINT##',0
0163'   4E 54 23 23
0167'   00
0168'                   lval:     DS      2
016A'   5E              getmem::LD        E,(HL)
016B'   23                INC     HL
016C'   56                LD      D,(HL)  ;then DE means sym=lval[0]
016D'   1A                LD      A,(DE)
016E'   FE 03             CP      3
0170'   28 24             JR      Z,CC22  ;if(sym[IDENT] != POINTER)
0172'   21 0001           LD      HL,1
0175'   19                ADD     HL,DE
0176'   7E                LD      A,(HL)
0177'   FE 04             CP      4
0179'   20 1B             JR      NZ,CC22 ;;if(sym[TYPE]==CCHAR)
017B'   21 01AB'          LD      HL,CC21
017E'   D5                PUSH    DE
017F'   CD 0C40'          CALL    ot
0182'   D1                POP     DE
0183'   21 0008           LD      HL,8
0186'   19                ADD     HL,DE
0187'   CD 0C40'          CALL    ot
018A'   21 01B3'          LD      HL,CC21+8
018D'   CD 0BF9'          CALL    ol
0190'   21 01B5'          LD      HL,CC21+10
0193'   C3 03D8'          JP      ffcall
0196'   21 01BD'        CC22:     LD      HL,CC21+12H
0199'   D5                PUSH    DE
019A'   CD 0C40'          CALL    ot
019D'   D1                POP     DE
019E'   21 0008           LD      HL,8
01A1'   19                ADD     HL,DE
01A2'   CD 0C40'          CALL    ot
01A5'   21 01C6'          LD      HL,CC21+1BH
01A8'   C3 0BF9'          JP      ol
01AB'   20 4C 44 20 CC21:         DB      ' LD A,(',0
01AF'   41 2C 28 00
01B3'   29 00             DB      ')',0
01B5'   43 43 53 58       DB      'CCSXT##',0
01B9'   54 23 23 00
01BD'   20 4C 44 20       DB      ' LD HL,(',0
01C1'   48 4C 2C 28
01C5'   00
01C6'   29 00             DB      ')',0
01C8'                   getloc::
01C8'   11 0006           LD      DE,6
01CB'   19                ADD     HL,DE   ;hl = sym + OFFSET
01CC'   7E                LD      A,(HL)
01CD'   23                INC     HL
01CE'   66                LD      H,(HL)
01CF'   6F                LD      L,A
01D0'   ED 5B 0000*       LD      DE,(csp)
01D4'   A7                AND     A
01D5'   ED 52             SBC     HL,DE
01D7'   E5                PUSH    HL
01D8'   CD 0000*          CALL    const
01DB'   C1                POP     BC
01DC'   21 01E2'          LD      HL,CC24
01DF'   C3 0BF9'          JP      ol
01E2'   20 41 44 44 CC24:         DB      ' ADD HL,SP',0
01E6'   20 48 4C 2C
01EA'   53 50 00
01ED'   C1              putmem::POP       BC
01EE'   E1                POP     HL
01EF'   E5                PUSH    HL
01F0'   C5                PUSH    BC
01F1'   5E                LD      E,(HL)
01F2'   23                INC     HL
01F3'   56                LD      D,(HL)
01F4'   1A                LD      A,(DE)
01F5'   FE 03             CP      3
01F7'   28 1E             JR      Z,CC26
01F9'   21 0001           LD      HL,1
01FC'   19                ADD     HL,DE
01FD'   7E                LD      A,(HL)
01FE'   FE 04             CP      4
0200'   20 15             JR      NZ,CC26
0202'   21 022C'          LD      HL,CC25
0205'   D5                PUSH    DE
0206'   CD 0C40'          CALL    ot
0209'   D1                POP     DE
020A'   21 0008           LD      HL,8
020D'   19                ADD     HL,DE
020E'   CD 0C40'          CALL    ot
0211'   21 023A'          LD      HL,CC25+0EH
0214'   C3 0BF9'          JP      ol
0217'   21 023E'        CC26:     LD      HL,CC25+12H
021A'   D5                PUSH    DE
021B'   CD 0C40'          CALL    ot
021E'   D1                POP     DE
021F'   21 0008           LD      HL,8
0222'   19                ADD     HL,DE
0223'   CD 0C40'          CALL    ot
0226'   21 0244'          LD      HL,CC25+16H
0229'   C3 0BF9'          JP      ol
022C'   20 4C 44 20 CC25:         DB      ' LD A,L',CR
0230'   41 2C 4C 0D
0234'   20 4C 44 20       DB      ' LD (',0
0238'   28 00
023A'   29 2C 41 00       DB      '),A',0
023E'   20 4C 44 20       DB      ' LD (',0
0242'   28 00
0244'   29 2C 48 4C       DB      '),HL',0
0248'   00
0249'   22 0168'        putstk::LD        (lval),HL
024C'   3A 0000*          LD      A,(explevel)
024F'   A7                AND     A
0250'   CA 0265'          JP      Z,CC30
0253'   5E                LD      E,(HL)
0254'   23                INC     HL
0255'   56                LD      D,(HL)  ;DE = sym = lval[0]
0256'   23                INC     HL
0257'   23                INC     HL
0258'   13                INC     DE
0259'   13                INC     DE
```

▶黄金週間PRO-68Kは「これでフォーマットできまい。まいったか」という編集部の執念が伝わってくる。
佐藤　研一(21)茨城県

```
025A'  13           INC   DE
025B'  13           INC   DE
025C'  1A           LD    A,(DE)
025D'  77           LD    (HL),A
025E'  23           INC   HL
025F'  36 00        LD    (HL),0  ;lval[1] = sym[ITYPE]
0261'  AF           XOR   A
0262'  32 0000*     LD    (explevel),A
0265'  2A 0168'  CC30:   LD    HL,(lval)
0268'  23           INC   HL
0269'  23           INC   HL
026A'  7E           LD    A,(HL)
026B'  FE 04        CP    4
026D'  21 0279'     LD    HL,CC28
0270'  CA 0BF9'     JP    Z,ol
0273'  21 028C'     LD    HL,CC28+13H
0276'  C3 03D8'     JP    ffcall
0279'  20 4C 44 20 CC28:   DB    ' LD A,L',CR
027D'  41 2C 4C 0D
0281'  20 4C 44 20  DB    ' LD (DE),A',0
0285'  28 44 45 29
0289'  2C 41 00
028C'  43 43 50 49  DB    'CCPINT##',0
0290'  4E 54 23 23
0294'  00
0295'  21 029B'  move::  LD    HL,CC31
0298'  C3 0BF9'     JP    ol
029B'  20 4C 44 20 CC31:   DB    ' LD D,H',CR
029F'  44 2C 48 0D
02A3'  20 4C 44 20  DB    ' LD E,L',0
02A7'  45 2C 4C 00
02AB'            swap::
02AB'  21 02B1'  swap2::  LD    HL,CC32
02AE'  C3 0BF9'     JP    ol
02B1'  20 45 58 20 CC32:   DB    ' EX DE,HL',0
02B5'  44 45 2C 48
02B9'  4C 00
02BB'  21 02C1'  immed::  LD    HL,CC33
02BE'  C3 0C40'     JP    ot
02C1'  20 4C 44 20 CC33:   DB    ' LD HL,',0
02C5'  48 4C 2C 00
02C9'  21 02CF'  immed2::LD    HL,CC34
02CC'  C3 0C40'     JP    ot
02CF'  20 4C 44 20 CC34:   DB    ' LD DE,',0
02D3'  44 45 2C 00
02D7'  21 02E6'  push::  LD    HL,CC35
02DA'  CD 0BF9'     CALL  ol
02DD'  2A 0000*     LD    HL,(csp)
02E0'  2B           DEC   HL
02E1'  2B           DEC   HL
02E2'  22 0000*     LD    (csp),HL
02E5'  C9           RET
02E6'  20 50 55 53 CC35:   DB    ' PUSH HL;',0
02EA'  48 20 48 4C
02EE'  3B 00
02F0'            smartpop::
02F0'  C1           POP   BC
02F1'  D1           POP   DE
02F2'  E1           POP   HL
02F3'  E5           PUSH  HL
02F4'  D5           PUSH  DE
02F5'  C5           PUSH  BC
02F6'  ED 53 030E'  LD    (Sstart),DE
02FA'  11 000A      LD    DE,10
02FD'  19           ADD   HL,DE
02FE'  7E           LD    A,(HL)
02FF'  23           INC   HL
0300'  66           LD    H,(HL)
0301'  6F           LD    L,A
0302'  B4           OR    H
0303'  C2 039E'     JP    NZ,pop
0306'  ED 5B 030E'  LD    DE,(Sstart)
030A'  CD 0310'     CALL  unpush
030D'  C9           RET
030E'            Sstart:  DS    2
0310'  21 037D'  unpush: LD    HL,CC39
0313'  7E        CC40:   LD    A,(HL)
0314'  A7           AND   A
0315'  28 05        JR    Z,CC41
0317'  12           LD    (DE),A
0318'  23           INC   HL
0319'  13           INC   DE
031A'  18 F7        JR    CC40
031C'  ED 53 0392' CC41:   LD    (Udest),DE
0320'  2A 0000*     LD    HL,(stagenex)
0323'  22 0394'     LD    (Usour),HL
0326'  ED 5B 0394' CC42:   LD    DE,(Usour)
032A'  1B           DEC   DE
032B'  ED 53 0394'  LD    (Usour),DE
032F'  2A 0392'     LD    HL,(Udest)
0332'  A7           AND   A
0333'  ED 52        SBC   HL,DE
0335'  30 3D        JR    NC,CC46
0337'  2A 0394'     LD    HL,(Usour)
033A'  11 0387'     LD    DE,CC39+0AH
033D'  CD 0BF0'     CALL  _streq
0340'  20 E4        JR    NZ,CC42
0342'  2A 0394'     LD    HL,(Usour)      ;--sour
0345'  2B           DEC   HL
0346'  22 0394'     LD    (Usour),HL
0349'  3E 02        LD    A,2             ;i=BPW
034B'  32 0396'     LD    (Ui),A
034E'  2A 0394'  CC45:   LD HL,(Usour) ;if(ishex(*(--sour)))
0351'  2B           DEC   HL
0352'  22 0394'     LD    (Usour),HL
0355'  7E           LD    A,(HL)
0356'  CD 0397'     CALL  isdigit
0359'  38 CB        JR    C,CC42
035B'  3A 0396'     LD    A,(Ui)
035E'  47           LD    B,A
035F'  7E           LD    A,(HL)
0360'  90           SUB   B
0361'  77           LD    (HL),A  ;*sour = *sour-i
0362'  FE 30        CP    '0'
0364'  30 07        JR    NC,CC47
0366'  C6 0A        ADD   A,10    ;*sour = *sour+10
0368'  77           LD    (HL),A
0369'  3E 01        LD    A,1             ;i=1
036B'  18 01        JR    CC48
036D'  AF        CC47:   XOR   A       ;i=0
036E'  32 0396'  CC48:   LD    (Ui),A
0371'  C3 034E'     JP    CC45
0374'  2A 0000*  CC46:   LD    HL,(csp)
0377'  23           INC   HL
0378'  23           INC   HL
0379'  22 0000*     LD    (csp),HL
037C'  C9           RET
037D'  20 45 58 20 CC39:   DB    ' EX DE,HL',0
0381'  44 45 2C 48
0385'  4C 00
0387'  20 41 44 44  DB    ' ADD HL,SP',0
038B'  20 48 4C 2C
038F'  53 50 00
0392'            Udest:   DS    2
0394'            Usour:   DS    2
0396'            Ui:      DS    1
0397'  FE 30     isdigit:CP    '0'
0399'  D8           RET   C
039A'  FE 3A        CP    '9'+1
039C'  3F           CCF
039D'  C9           RET
039E'  21 03AD'  pop::   LD    HL,CC58
03A1'  CD 0BF9'     CALL  ol
03A4'  2A 0000*     LD    HL,(csp)
03A7'  23           INC   HL
03A8'  23           INC   HL
03A9'  22 0000*     LD    (csp),HL
03AC'  C9           RET
03AD'  20 50 4F 50 CC58:   DB    ' POP DE',0
03B1'  20 44 45 00
03B5'            swapstk::
03B5'  21 03BB'     LD    HL,CC59
03B8'  C3 0BF9'     JP    ol
03BB'  20 45 58 20 CC59:   DB    ' EX (SP),HL',0
03BF'  28 53 50 29
03C3'  2C 48 4C 00
03C7'  21 03CD'  sw::    LD    HL,CC60
03CA'  C3 03D8'     JP    ffcall
03CD'  43 43 53 57 CC60:   DB    'CCSWITCH##',0
03D1'  49 54 43 48
03D5'  23 23 00
03D8'  E5        ffcall::PUSH  HL
03D9'  21 03E3'     LD    HL,CC61
03DC'  CD 0C40'     CALL  ot
03DF'  E1           POP   HL
03E0'  C3 0BF9'     JP    ol
03E3'  20 43 41 4C CC61:   DB    ' CALL ',0
03E7'  4C 20 00
03EA'  21 03F0'  ffret:: LD    HL,CC62
03ED'  C3 0BF9'     JP    ol
03F0'  20 52 45 54 CC62:   DB    ' RET',0
03F4'  00
03F5'            callstk::
03F5'  21 03FB'     LD    HL,CC63
03F8'  C3 03D8'     JP    ffcall
03FB'  43 43 44 43 CC63:   DB    'CCDCAL##',0
03FF'  41 4C 23 23
0403'  00
0404'            ult0::
0404'  21 0414'  jump::  LD    HL,CC64
0407'  CD 0C40'     CALL  ot
040A'  C1           POP   BC
040B'  E1           POP   HL
040C'  E5           PUSH  HL
040D'  C5           PUSH  BC
040E'  CD 0C05'     CALL  printlab
0411'  C3 0000*     JP    nl
0414'  20 4A 50 20 CC64:   DB    ' JP ',0
0418'  00
0419'            zerojump::
0419'  C1           POP   BC
041A'  E1           POP   HL
041B'  E5           PUSH  HL
041C'  C5           PUSH  BC
041D'  11 000E      LD    DE,0EH
0420'  19           ADD   HL,DE
0421'  CD 0000*     CALL  CCGINT##
0424'  E5           PUSH  HL
0425'  21 0000      LD    HL,00H
0428'  E5           PUSH  HL
0429'  CD 0000*     CALL  clearsta
042C'  C1           POP   BC
042D'  C1           POP   BC
042E'  21 0006      LD    HL,06H
0431'  39           ADD   HL,SP
0432'  CD 0000*     CALL  CCGINT##
0435'  E5           PUSH  HL
0436'  21 0006      LD    HL,06H
0439'  39           ADD   HL,SP
043A'  CD 0000*     CALL  CCGINT##
043D'  E3           EX    (SP),HL
043E'  CD 0000*     CALL  CCDCAL##
0441'  C1           POP   BC
0442'  C9           RET
0443'            defstora::
0443'  11 0001      LD    DE,1
0446'  A7           AND   A
0447'  ED 52        SBC   HL,DE
0449'  21 0455'     LD    HL,CC67
044C'  CA 0C40'     JP    Z,ot
044F'  21 045A'     LD    HL,CC67+05H
0452'  C3 0C40'     JP    ot
0455'  20 44 42 20 CC67:   DB    ' DB ',0
0459'  00
045A'  20 44 57 20  DB    ' DW ',0
045E'  00
045F'  21 0465'  point:: LD    HL,CC70
0462'  C3 0BF9'     JP    ol
0465'  20 44 57 20 CC70:   DB    ' DW $+2',0
0469'  24 2B 32 00
046D'  C1        modstk::POP   BC
046E'  D1           POP   DE
046F'  E1           POP   HL
0470'  E5           PUSH  HL
0471'  D5           PUSH  DE
0472'  C5           PUSH  BC
0473'  22 054F'     LD    (Mnewsp),HL
0476'  ED 53 0551'  LD    (Msave),DE
047A'  ED 5B 0000*  LD    DE,(csp)
047E'  A7           AND   A
047F'  ED 52        SBC   HL,DE
0481'  22 0553'     LD    (Mk),HL ; k = newsp - csp
0484'  7C           LD    A,H     ;if(k==0) return(newsp)
0485'  B5           OR    L
0486'  CA 0515'     JP    Z,CC84
0489'  AF           XOR   A
048A'  B4           OR    H
048B'  FA 04BD'     JP    M,CC73  ;if(k>0)  {
048E'  11 0007      LD    DE,7    ;  if(k<7)  {
0491'  A7           AND   A
0492'  ED 52        SBC   HL,DE
0494'  30 27        JR    NC,CC73
0496'  3A 0553'     LD    A,(Mk)  ;    if(k&1) {
0499'  E6 01        AND   1
049B'  28 0A        JR    Z,CC75
049D'  21 0519'     LD    HL,CC71 ;      ol("INC SP")
04A0'  CD 0BF9'     CALL  ol
04A3'  21 0553'     LD    HL,Mk   ;      k--
04A6'  35           DEC   (HL)    ;    }
04A7'  3A 0553'  CC75:   LD A,(Mk) ;   while(k)  {
04AA'  A7           AND   A
04AB'  28 68        JR    Z,CC84
04AD'  21 0521'     LD    HL,CC71+08H ;     ol("POP BC")
04B0'  CD 0BF9'     CALL  ol
04B3'  3A 0553'     LD    A,(Mk)      ;     k-=2
04B6'  3D           DEC   A
04B7'  3D           DEC   A               ;  }
04B8'  32 0553'     LD    (Mk),A
04BB'  18 EA        JR    CC75



04BD'  2A 0553'  CC73:   LD    HL,(Mk) ;if(k<0)  {
04C0'  AF           XOR   A
04C1'  B4           OR    H
04C2'  F2 04F7'     JP    P,CC78
04C5'  11 FFF9      LD    DE,-7   ;  if(k<-7)  {
04C8'  A7           AND   A
04C9'  ED 52        SBC   HL,DE
04CB'  38 2A        JR    C,CC79
04CD'  3A 0553'     LD    A,(Mk)  ;    if(k&1)  {
04D0'  E6 01        AND   1
04D2'  28 0D        JR    Z,CC80
04D4'  21 0529'     LD    HL,CC71+10H ; ol("DEC SP")
04D7'  CD 0BF9'     CALL  ol
04DA'  3A 0553'     LD    A,(Mk)  ;       k++
04DD'  3C           INC   A
04DE'  32 0553'     LD    (Mk),A  ;     }
04E1'  3A 0553'  CC80:   LD    A,(Mk)  ;while(k) {
04E4'  A7           AND   A
04E5'  28 2E        JR    Z,CC84
04E7'  21 0531'     LD    HL,CC71+18H ; ol("PUSH BC");
04EA'  CD 0BF9'     CALL  ol
04ED'  3A 0553'     LD    A,(Mk)  ;  k = k + BPW;
04F0'  3C           INC   A
04F1'  3C           INC   A
04F2'  32 0553'     LD    (Mk),A
04F5'  18 EA        JR    CC80
04F7'            CC79:
04F7'            CC78:
04F7'  2A 0551'     LD    HL,(Msave)
04FA'  7C           LD    A,H
04FB'  B5           OR    L
04FC'  C4 02AB'     CALL  NZ,swap
04FF'  2A 0553'     LD    HL,(Mk)
0502'  E5           PUSH  HL
0503'  CD 0000*     CALL  const
0506'  C1           POP   BC
0507'  21 053A'     LD    HL,CC71+21H
050A'  CD 0BF9'     CALL  ol
050D'  2A 0551'     LD    HL,(Msave)
0510'  7C           LD    A,H
0511'  B5           OR    L
0512'  C4 02AB'     CALL  NZ,swap
0515'  2A 054F'  CC84:   LD    HL,(Mnewsp)
0518'  C9           RET
0519'  20 49 4E 43 CC71:   DB    ' INC SP',0
051D'  20 53 50 00
0521'  20 50 4F 50  DB    ' POP BC',0
0525'  20 42 43 00
0529'  20 44 45 43  DB    ' DEC SP',0
052D'  20 53 50 00
0531'  20 50 55 53  DB    ' PUSH BC',0
0535'  48 20 42 43
0539'  00
053A'  20 41 44 44  DB    ' ADD HL,SP',CR
053E'  20 48 4C 2C
0542'  53 50 0D
0545'  20 4C 44 20  DB    ' LD SP,HL',0
0549'  53 50 2C 48
054D'  4C 00
054F'            Mnewsp:  DS    2
0551'            Msave:   DS    2
0553'            Mk:      DS    2
0555'            doublere::
0555'  21 055B'     LD    HL,CC85
0558'  C3 0BF9'     JP    ol
055B'  20 41 44 44 CC85:   DB    ' ADD HL,HL',0
055F'  20 48 4C 2C
0563'  48 4C 00
0566'  21 056C'  ffadd:: LD    HL,CC86
0569'  C3 0BF9'     JP    ol
056C'  20 41 44 44 CC86:   DB    ' ADD HL,DE',0
0570'  20 48 4C 2C
0574'  44 45 00
0577'  21 057D'  ffsub:: LD    HL,CC87
057A'  C3 03D8'     JP    ffcall
057D'  43 43 53 55 CC87:   DB    'CCSUB##',0
0581'  42 23 23 00
0585'  21 058B'  ffmult::LD    HL,CC88
0588'  C3 03D8'     JP    ffcall
058B'  43 43 4D 55 CC88:   DB    'CCMULT##',0
058F'  4C 54 23 23
0593'  00
0594'  21 059A'  ffdiv:: LD    HL,CC89
0597'  C3 03D8'     JP    ffcall
059A'  43 43 44 49 CC89:   DB    'CCDIV##',0
059E'  56 23 23 00
05A2'  CD 0594'  ffmod:: CALL  ffdiv
05A5'  C3 02AB'     JP    swap
05A8'  21 05AE'  ffor::  LD    HL,CC91
05AB'  C3 03D8'     JP    ffcall
05AE'  43 43 4F 52 CC91:   DB    'CCOR##',0
05B2'  23 23 00
05B5'  21 05BB'  ffxor:: LD    HL,CC92
05B8'  C3 03D8'     JP    ffcall
05BB'  43 43 58 4F CC92:   DB    'CCXOR##',0
05BF'  52 23 23 00
05C3'  21 05C9'  ffand:: LD    HL,CC93
05C6'  C3 03D8'     JP    ffcall
05C9'  43 43 41 4E CC93:   DB    'CCAND##',0
05CD'  44 23 23 00
05D1'  21 05D7'  lneg::  LD    HL,CC94
05D4'  C3 03D8'     JP    ffcall
05D7'  43 43 4C 4E CC94:   DB    'CCLNEG##',0
05DB'  45 47 23 23
05DF'  00
05E0'  21 05E6'  ffasr:: LD    HL,CC95
05E3'  C3 03D8'     JP    ffcall
05E6'  43 43 41 53 CC95:   DB    'CCASR##',0
05EA'  52 23 23 00
05EE'  21 05F4'  ffasl:: LD    HL,CC96
05F1'  C3 03D8'     JP    ffcall
05F4'  43 43 41 53 CC96:   DB    'CCASL##',0
05F8'  4C 23 23 00
05FC'  21 0602'  neg::   LD    HL,CC97
05FF'  C3 03D8'     JP    ffcall
0602'  43 43 4E 45 CC97:   DB    'CCNEG##',0
0606'  47 23 23 00
060A'  21 0610'  com::   LD    HL,CC98
060D'  C3 03D8'     JP    ffcall
0610'  43 43 43 4F CC98:   DB    'CCCOM##',0
0614'  4D 23 23 00
0618'  CD 02D7'  getmd:: CALL  push
061B'  CD 02EB'     CALL  immed
061E'  C1           POP   BC
061F'  E1           POP   HL
0620'  E5           PUSH  HL
0621'  C5           PUSH  BC
0622'  E5           PUSH  HL
0623'  CD 0000*     CALL  outdec
0626'  C1           POP   BC
0627'  CD 0000*     CALL  nl
062A'  CD 0585'     CALL  ffmult
062D'  C3 039E'     JP    pop
0630'  C1        inc::   POP   BC
0631'  E1           POP   HL
0632'  E5           PUSH  HL
0633'  C5           PUSH  BC
0634'  E5        incl:   PUSH  HL
0635'  21 0645'     LD    HL,CC99
0638'  CD 0BF9'     CALL  ol
063B'  E1           POP   HL
063C'  7C           LD    A,H
063D'  B5           OR    L
063E'  C8           RET   Z
063F'  2B           DEC   HL
0640'  7C           LD    A,H
0641'  B5           OR    L
0642'  20 F0        JR    NZ,incl
0644'  C9           RET
0645'  20 49 4E 43 CC99:   DB    ' INC HL',0
0649'  20 48 4C 00
064D'  C1        dec::   POP   BC
064E'  E1           POP   HL
064F'  E5           PUSH  HL
0650'  C5           PUSH  BC
0651'  E5        decl:   PUSH  HL
0652'  21 0662'     LD    HL,CC103
0655'  CD 0BF9'     CALL  ol
0658'  E1           POP   HL
0659'  7C           LD    A,H
065A'  B5           OR    L
065B'  C8           RET   Z
065C'  2B           DEC   HL
065D'  7C           LD    A,H
065E'  B5           OR    L
065F'  20 F0        JR    NZ,decl
0661'  C9           RET
0662'  20 44 45 43 CC103:  DB    ' DEC HL',0
0666'  20 48 4C 00
066A'  21 0670'  ffeq::  LD    HL,CC107
066D'  C3 03D8'     JP    ffcall
0670'  43 43 45 51 CC107:  DB    'CCEQ##',0
0674'  23 23 00
0677'  21 0687'  eq0::   LD    HL,CC108
067A'  CD 0C40'     CALL  ot
067D'  C1           POP   BC
067E'  E1           POP   HL
067F'  E5           PUSH  HL
0680'  C5           PUSH  BC
0681'  CD 0C05'     CALL  printlab
0684'  C3 0000*     JP    nl
0687'  20 4C 44 20 CC108:  DB    ' LD A,H',CR
068B'  41 2C 48 0D
068F'  20 4F 52 20  DB    ' OR L',CR
0693'  4C 0D
0695'  20 4A 50 20  DB    ' JP NZ,',0
0699'  4E 5A 2C 00
069D'  21 06A3'  ffne::  LD    HL,CC109
06A0'  C3 03D8'     JP    ffcall
06A3'  43 43 4E 45 CC109:  DB    'CCNE##',0
06A7'  23 23 00
06AA'            testjump::
06AA'  21 06BA'  ne0::   LD    HL,CC110
06AD'  CD 0C40'     CALL  ot
06B0'  C1           POP   BC
06B1'  E1           POP   HL
06B2'  E5           PUSH  HL
06B3'  C5           PUSH  BC
06B4'  CD 0C05'     CALL  printlab
06B7'  C3 0000*     JP    nl
06BA'  20 4C 44 20 CC110:  DB    ' LD A,H',CR
06BE'  41 2C 48 0D
06C2'  20 4F 52 20  DB    ' OR L',CR
06C6'  4C 0D
06C8'  20 4A 50 20  DB    ' JP Z,',0
06CC'  5A 2C 00
06CF'  21 06D5'  fflt::  LD    HL,CC111
06D2'  C3 03D8'     JP    ffcall
06D5'  43 43 4C 54 CC111:  DB    'CCLT##',0
06D9'  23 23 00
06DC'  21 06EC'  lt0::   LD    HL,CC112
06DF'  CD 0C40'     CALL  ot
06E2'  C1           POP   BC
06E3'  E1           POP   HL
06E4'  E5           PUSH  HL
06E5'  C5           PUSH  BC
06E6'  CD 0C05'     CALL  printlab
06E9'  C3 0000*     JP    nl
06EC'  20 58 4F 52 CC112:  DB    ' XOR A',CR
06F0'  20 41 0D
06F3'  20 4F 52 20  DB    ' OR H',CR
06F7'  48 0D
06F9'  20 4A 50 20  DB    ' JP P,',0
06FD'  50 2C 00
0700'  21 0706'  ffle::  LD    HL,CC113
0703'  C3 03D8'     JP    ffcall
0706'  43 43 4C 45 CC113:  DB    'CCLE##',0
070A'  23 23 00
070D'  21 071D'  le0::   LD    HL,CC114
0710'  CD 0C40'     CALL  ot
0713'  C1           POP   BC
0714'  E1           POP   HL
0715'  E5           PUSH  HL
0716'  C5           PUSH  BC
0717'  CD 0C05'     CALL  printlab
```

▶付録ディスクのfloat2＋の「versin」というのが意味ありげで気になってしょうがない。こんな僕はちょっと変？
石渡　貴史(19)神奈川県

```
071A'  C3 0000*          JP      nl
071D'  20 4C 44 20 CC114: DB      ' LD A,H',CR
0721'  41 2C 48 0D
0725'  20 4F 52 20   DB    ' OR L',CR
0729'  4C 0D
072B'  20 4A 52 20   DB    ' JR Z,$+8',CR
072F'  5A 2C 24 2B
0733'  38 0D
0735'  20 58 4F 52   DB    ' XOR A',CR
0739'  20 41 0D
073C'  20 4F 52 20   DB    ' OR H',CR
0740'  48 0D
0742'  20 4A 50 20   DB    ' JP P,',0
0746'  50 2C 00
0749'  21 074F'    ffgt::  LD      HL,CC115
074C'  C3 03D8'      JP    ffcall
074F'  43 43 47 54 CC115: DB      'CCGT##',0
0753'  23 23 00
0756'  21 0776'    gt0::   LD      HL,CC116
0759'  CD 0C40'      CALL  ot
075C'  C1            POP   BC
075D'  E1            POP   HL
075E'  E5            PUSH  HL
075F'  C5            PUSH  BC
0760'  CD 0C05'      CALL  printlab
0763'  CD 0000*      CALL  nl
0766'  21 078A'      LD    HL,CC116+14H
0769'  CD 0C40'      CALL  ot
076C'  C1            POP   BC
076D'  E1            POP   HL
076E'  E5            PUSH  HL
076F'  C5            PUSH  BC
0770'  CD 0C05'      CALL  printlab
0773'  C3 0000*      JP    nl
0776'  20 58 4F 52 CC116: DB      ' XOR A',CR
077A'  20 41 0D
077D'  20 4F 52 20   DB    ' OR H',CR
0781'  48 0D
0783'  20 4A 50 20   DB    ' JP M,',0
0787'  4D 2C 00
078A'  20 4F 52 20   DB    ' OR L',CR
078E'  4C 0D
0790'  20 4A 50 20   DB    ' JP Z,',0
0794'  5A 2C 00
0797'  21 079D'    ffge::  LD      HL,CC117
079A'  C3 03D8'      JP    ffcall
079D'  43 43 47 45 CC117: DB      'CCGE##',0
07A1'  23 23 00
07A4'  21 07B4'    ge0::   LD      HL,CC118
07A7'  CD 0C40'      CALL  ot
07AA'  C1            POP   BC
07AB'  E1            POP   HL
07AC'  E5            PUSH  HL
07AD'  C5            PUSH  BC
07AE'  CD 0C05'      CALL  printlab
07B1'  C3 0000*      JP    nl
07B4'  20 58 4F 52 CC118: DB      ' XOR A',CR
07B8'  20 41 0D
07BB'  20 4F 52 20   DB    ' OR H',CR
07BF'  48 0D
07C1'  20 4A 50 20   DB    ' JP M,',0
07C5'  4D 2C 00
07C6'  21 07CE'    ult::   LD      HL,CC119
07CB'  C3 03D8'      JP    ffcall
07CE'  43 43 55 4C CC119: DB      'CCULT##',0
07D2'  54 23 23 00
07D6'  21 07DC'    ule::   LD      HL,CC121
07D9'  C3 03D8'      JP    ffcall
07DC'  43 43 55 4C CC121: DB      'CCULE##',0
07E0'  45 23 23 00
07E4'  21 07EA'    ugt::   LD      HL,CC122
07E7'  C3 03D8'      JP    ffcall
07EA'  43 43 55 47 CC122: DB      'CCUGT##',0
07EE'  54 23 23 00
07F2'  21 07F8'    uge::   LD      HL,CC123
07F5'  C3 03D8'      JP    ffcall
07F8'  43 43 55 47 CC123: DB      'CCUGE##',0
07FC'  45 23 23 00
0800'              peephole::
0800'  22 0BE8'      LD    (_ptr),HL
0803'  2A 0BE8'    peepl:  LD      HL,(_ptr)
0806'  7E            LD    A,(HL)
0807'  A7            AND   A
0808'  C8            RET   Z
0809'  CD 0825'      CALL  OPTMZ
080C'  38 F5         JR    c,peepl
080E'  2A 0BE8'      LD    HL,(_ptr)
0811'  7E            LD    A,(HL)
0812'  23            INC   HL
0813'  22 0BE8'      LD    (_ptr),HL
0816'  26 00         LD    H,0
0818'  6F            LD    L,A
0819'  E5            PUSH  HL
081A'  2A 0000*      LD    HL,(output)
081D'  E5            PUSH  HL
081E'  CD 0000*      CALL  cout
0821'  C1            POP   BC
0822'  C1            POP   BC
0823'  18 DE         JR    peepl
0825'  11 0857'    OPTMZ:  LD      DE,CC140
0828'  06 04         LD    B,4
082A'  3A 0000*      LD    A,(optimize)
082D'  A7            AND   A
082E'  28 02         JR    Z,peep4
0830'  06 0F         LD    B,15
0832'  2A 0BE8'    peep4:  LD      HL,(_ptr)
0835'  D5            PUSH  DE
0836'  CD 0BF0'      CALL  _streq
0839'  D1            POP   DE
083A'  28 0E         JR    Z,peep5
083C'  EB            EX    DE,HL
083D'  CD 0BEA'      CALL  _skip
0840'  23            INC   HL
0841'  CD 0BEA'      CALL  _skip
0844'  23            INC   HL
0845'  EB            EX    DE,HL
0846'  10 EA         DJNZ  peep4
0848'  A7            AND   A       ;cy = 0
0849'  C9            RET
084A'  22 0BE8'    peep5:  LD      (_ptr),HL
084D'  EB            EX    DE,HL
084E'  CD 0BEA'      CALL  _skip
0851'  23            INC   HL
0852'  CD 0BF9'      CALL  ol
0855'  37            SCF
0856'  C9            RET
0857'  20 4C 44 20 CC140: DB ' LD HL,0',CR ; 1
085B'  48 4C 2C 30
085F'  0D
0860'  20 41 44 44   DB    ' ADD HL,SP',CR
0864'  20 48 4C 2C
0868'  53 50 0D
086B'  20 43 41 4C   DB    ' CALL CCGINT##',CR
086F'  4C 20 43 43
0873'  47 49 4E 54
0877'  23 23 0D
087A'  20 45 58 20   DB    ' EX DE,HL',CR,0
087E'  44 45 2C 48
0882'  4C 0D 00
0885'  20 50 4F 50   DB    ' POP DE',CR
0889'  20 44 45 0D
088D'  20 50 55 53   DB    ' PUSH DE',0
0891'  48 20 44 45
0895'  00
0896'  20 4C 44 20   DB    ' LD HL,0',CR ; 2
089A'  48 4C 2C 30
089E'  0D
089F'  20 41 44 44   DB    ' ADD HL,SP',CR
08A3'  20 48 4C 2C
08A7'  53 50 0D
08AA'  20 43 41 4C   DB    ' CALL CCGINT##',CR,0
08AE'  4C 20 43 43
08B2'  47 49 4E 54
08B6'  23 23 0D 00
08BA'  20 50 4F 50   DB    ' POP HL',CR
08BE'  20 48 4C 0D
08C2'  20 50 55 53   DB    ' PUSH HL',0
08C6'  48 20 48 4C
08CA'  00
08CB'  20 4C 44 20   DB    ' LD HL,2',CR  ; 3
08CF'  48 4C 2C 32
08D3'  0D
08D4'  20 41 44 44   DB    ' ADD HL,SP',CR
08D8'  20 48 4C 2C
08DC'  53 50 0D
08DF'  20 43 41 4C   DB    ' CALL CCGINT##',CR
08E3'  4C 20 43 43
08E7'  47 49 4E 54
08EB'  23 23 0D
08EE'  20 45 58 20   DB    ' EX DE,HL',CR,0
08F2'  44 45 2C 48
08F6'  4C 0D 00
08F9'  20 50 4F 50   DB    ' POP BC',CR
08FD'  20 42 43 0D
0901'  20 50 4F 50   DB    ' POP DE',CR
0905'  20 44 45 0D
0909'  20 50 55 53   DB    ' PUSH DE',CR
090D'  48 20 44 45
0911'  0D
0912'  20 50 55 53   DB    ' PUSH BC',0
0916'  48 20 42 43
091A'  00
091B'  20 4C 44 20   DB    ' LD HL,2',CR   ; 4
091F'  48 4C 2C 32
0923'  0D
0924'  20 41 44 44   DB    ' ADD HL,SP',CR
0928'  20 48 4C 2C
092C'  53 50 0D
092F'  20 43 41 4C   DB    ' CALL CCGINT##',CR,0
0933'  4C 20 43 43
0937'  47 49 4E 54
093B'  23 23 0D 00
093F'  20 50 4F 50   DB    ' POP BC',CR
0943'  20 42 43 0D
0947'  20 50 4F 50   DB    ' POP HL',CR
094B'  20 48 4C 0D
094F'  20 50 55 53   DB    ' PUSH HL',CR
0953'  48 20 48 4C
0957'  0D
0958'  20 50 55 53   DB    ' PUSH BC',0
095C'  48 20 42 43
0960'  00
0961'  20 41 44 44   DB    ' ADD HL,SP',CR ; 5
0965'  20 48 4C 2C
0969'  53 50 0D
096C'  20 43 41 4C   DB    ' CALL CCGINT##',CR,0
0970'  4C 20 43 43
0974'  47 49 4E 54
0978'  23 23 0D 00
097C'  20 43 41 4C   DB    ' CALL CCDSGI##',0
0980'  4C 20 43 43
0984'  44 53 47 49
0988'  23 23 00
098B'  20 41 44 44   DB    ' ADD HL,DE',CR ; 6
098F'  20 48 4C 2C
0993'  44 45 0D
0996'  20 43 41 4C   DB    ' CALL CCGINT##',CR,0
099A'  4C 20 43 43
099E'  47 49 4E 54
09A2'  23 23 0D 00
09A6'  20 43 41 4C   DB    ' CALL CCDDGI##',0
09AA'  4C 20 43 43
09AE'  44 44 47 49
09B2'  23 23 00
09B5'  20 41 44 44   DB    ' ADD HL,SP',CR ; 7
09B9'  20 48 4C 2C
09BD'  53 50 0D
09C0'  20 43 41 4C   DB    ' CALL CCGCHAR##',CR,0
09C4'  4C 20 43 43
09C8'  47 43 48 41
09CC'  52 23 23 0D
09D0'  00
09D1'  20 43 41 4C   DB    ' CALL CCDSGC##',0
09D5'  4C 20 43 43
09D9'  44 53 47 43
09DD'  23 23 00
09E0'  20 41 44 44   DB    ' ADD HL,DE',CR ; 8
09E4'  20 48 4C 2C
09E8'  44 45 0D
09EB'  20 43 41 4C   DB    ' CALL CCGCHAR##',CR,0
09EF'  4C 20 43 43
09F3'  47 43 48 41
09F7'  52 23 23 0D
09FB'  00
09FC'  20 43 41 4C   DB    ' CALL CCDDGC##',0
0A00'  4C 20 43 43
0A04'  44 44 47 43
0A08'  23 23 00
0A0B'  20 41 44 44   DB    ' ADD HL,SP',CR ; 9
0A0F'  20 48 4C 2C
0A13'  53 50 0D
0A16'  20 4C 44 20   DB    ' LD D,H',CR
0A1A'  44 2C 48 0D
0A1E'  20 4C 44 20   DB    ' LD E,L',CR
0A22'  45 2C 4C 0D
0A26'  20 43 41 4C   DB    ' CALL CCGINT##',CR
0A2A'  4C 20 43 43
0A2E'  47 49 4E 54
0A32'  23 23 0D
0A35'  20 49 4E 43   DB    ' INC HL',CR
0A39'  20 48 4C 0D
0A3D'  20 43 41 4C   DB    ' CALL CCPINT##',CR,0
0A41'  4C 20 43 43
0A45'  50 49 4E 54
0A49'  23 23 0D 00
0A4D'  20 43 41 4C   DB    ' CALL CCINCI##',0
0A51'  4C 20 43 43
0A55'  49 4E 43 49
0A59'  23 23 00
0A5C'  20 41 44 44   DB    ' ADD HL,SP',CR ; 10
0A60'  20 48 4C 2C
0A64'  53 50 0D
0A67'  20 4C 44 20   DB    ' LD D,H',CR
0A6B'  44 2C 48 0D
0A6F'  20 4C 44 20   DB    ' LD E,L',CR
0A73'  45 2C 4C 0D
0A77'  20 43 41 4C   DB    ' CALL CCGINT##',CR
0A7B'  4C 20 43 43
0A7F'  47 49 4E 54
0A83'  23 23 0D
0A86'  20 44 45 43   DB    ' DEC HL',CR
0A8A'  20 48 4C 0D
0A8E'  20 43 41 4C   DB    ' CALL CCPINT##',CR,0
0A92'  4C 20 43 43
0A96'  50 49 4E 54
0A9A'  23 23 0D 00
0A9E'  20 43 41 4C   DB    ' CALL CCDECI##',0
0AA2'  4C 20 43 43
0AA6'  44 45 43 49
0AAA'  23 23 00
0AAD'  20 41 44 44   DB    ' ADD HL,SP',CR ; 11
0AB1'  20 48 4C 2C
0AB5'  53 50 0D
0AB8'  20 4C 44 20   DB    ' LD D,H',CR
0ABC'  44 2C 48 0D
0AC0'  20 4C 44 20   DB    ' LD E,L',CR
0AC4'  45 2C 4C 0D
0AC8'  20 43 41 4C   DB    ' CALL CCGCHAR##',CR
0ACC'  4C 20 43 43
0AD0'  47 43 48 41
0AD4'  52 23 23 0D
0AD8'  20 49 4E 43   DB    ' INC HL',CR
0ADC'  20 48 4C 0D
0AE0'  20 4C 44 20   DB    ' LD A,L',CR
0AE4'  41 2C 4C 0D
0AE8'  20 4C 44 20   DB    ' LD (DE),A',CR,0
0AEC'  28 44 45 29
0AF0'  2C 41 0D 00
0AF4'  20 43 41 4C   DB    ' CALL CCINCC##',0
0AF8'  4C 20 43 43
0AFC'  49 4E 43 43
0B00'  23 23 00
0B03'  20 41 44 44   DB    ' ADD HL,SP',CR ; 12
0B07'  20 48 4C 2C
0B0B'  53 50 0D
0B0E'  20 4C 44 20   DB    ' LD D,H',CR
0B12'  44 2C 48 0D
0B16'  20 4C 44 20   DB    ' LD E,L',CR
0B1A'  45 2C 4C 0D
0B1E'  20 43 41 4C   DB    ' CALL CCGCHAR##',CR
0B22'  4C 20 43 43
0B26'  47 43 48 41
0B2A'  52 23 23 0D
0B2E'  20 44 45 43   DB    ' DEC HL',CR
0B32'  20 48 4C 0D
0B36'  20 4C 44 20   DB    ' LD A,L',CR
0B3A'  41 2C 4C 0D
0B3E'  20 4C 44 20   DB    ' LD (DE),A',CR,0
0B42'  28 44 45 29
0B46'  2C 41 0D 00
0B4A'  20 43 41 4C   DB    ' CALL CCDECC##',0
0B4E'  4C 20 43 43
0B52'  44 45 43 43
0B56'  23 23 00
0B59'  20 41 44 44   DB    ' ADD HL,DE',CR ; 13
0B5D'  20 48 4C 2C
0B61'  44 45 0D
0B64'  20 50 4F 50   DB    ' POP DE',CR
0B68'  20 44 45 0D
0B6C'  20 43 41 4C   DB    ' CALL CCPINT##',CR,0
0B70'  4C 20 43 43
0B74'  50 49 4E 54
0B78'  23 23 0D 00
0B7C'  20 43 41 4C   DB    ' CALL CDPDPI##',0
0B80'  4C 20 43 44
0B84'  50 44 50 49
0B88'  23 23 00
0B8B'  20 41 44 44   DB    ' ADD HL,DE',CR ; 14
0B8F'  20 48 4C 2C
0B93'  44 45 0D
0B96'  20 50 4F 50   DB    ' POP DE',CR
0B9A'  20 44 45 0D
0B9E'  20 4C 44 20   DB    ' LD A,L',CR
0BA2'  41 2C 4C 0D
0BA6'  20 4C 44 20   DB    ' LD (DE),A',CR,0
0BAA'  28 44 45 29
0BAE'  2C 41 0D 00
0BB2'  20 43 41 4C   DB    ' CALL CDPDPC##',0
0BB6'  4C 20 43 44
0BBA'  50 44 50 43
0BBE'  23 23 00
0BC1'  20 50 4F 50   DB    ' POP DE',CR    ; 15
0BC5'  20 44 45 0D
0BC9'  20 43 41 4C   DB    ' CALL CCPINT##',CR,0
0BCD'  4C 20 43 43
0BD1'  50 49 4E 54
0BD5'  23 23 0D 00
0BD9'  20 43 41 4C   DB    ' CALL CCPDPI##',0
0BDD'  4C 20 43 43
0BE1'  50 44 50 49
0BE5'  23 23 00
0BE8'              _ptr:   DS      2
0BEA'  7E          _skip:  LD      A,(HL)
0BEB'  A7            AND   A
0BEC'  C8            RET   Z
0BED'  23            INC   HL
0BEE'  18 FA         JR    _skip
0BF0'  1A          _streq: LD      A,(DE)
0BF1'  A7            AND   A
0BF2'  C8            RET   Z
0BF3'  BE            CP    (HL)
0BF4'  C0            RET   NZ
0BF5'  13            INC   DE
0BF6'  23            INC   HL
0BF7'  18 F7         JR    _streq
0BF9'  CD 0C40'    ol:     CALL    ot
0BFC'  21 000D       LD    HL,CR
0BFF'  E5            PUSH  HL
0C00'  CD 0000*      CALL  outbyte
0C03'  C1            POP   BC
0C04'  C9            RET
0C05'              printlab::
0C05'  E5            PUSH  HL
0C06'  21 0C29'      LD    HL,STRCC
0C09'  CD 0C40'      CALL  ot
0C0C'  E1            POP   HL
0C0D'  E5            PUSH  HL  ;Optimize ノ トキノタメニ
0C0E'  11 000A       LD    DE,10   ; 3 ケタ ニスル !!
0C11'  B7            OR    A
0C12'  ED 52         SBC   HL,DE
0C14'  DC 0C2C'      CALL  C,putzero
0C17'  E1            POP   HL
0C18'  E5            PUSH  HL
0C19'  11 0064       LD    DE,100
0C1C'  B7            OR    A
0C1D'  ED 52         SBC   HL,DE
0C1F'  DC 0C2C'      CALL  C,putzero
0C22'  E1            POP   HL
0C23'  E5            PUSH  HL
0C24'  CD 0000*      CALL  outdec
0C27'  C1            POP   BC
0C28'  C9            RET
0C29'  43 43 00    STRCC:  DB      'CC',0
0C2C'  E5          putzero:PUSH    HL
0C2D'  21 0030       LD    HL,'0'
0C30'  E5            PUSH  HL
0C31'  CD 0000*      CALL  outbyte
0C34'  E1            POP   HL
0C35'  E1            POP   HL
0C36'  C9            RET
0C37'              postlabe::
0C37'  CD 0C05'      CALL  printlab
0C3A'  CD 0000*      CALL  col
0C3D'  C3 0000*      JP    nl
0C40'  22 0C62'    ot:     LD      (Optr),HL
0C43'  21 0001       LD    HL,1
0C46'  E5            PUSH  HL
0C47'  CD 0000*      CALL  poll
0C4A'  C1            POP   BC
0C4B'  2A 0C62'    CC192:  LD      HL,(Optr)
0C4E'  7E            LD    A,(HL)
0C4F'  A7            AND   A
0C50'  C8            RET   Z
0C51'  26 00         LD    H,0
0C53'  6F            LD    L,A
0C54'  E5            PUSH  HL
0C55'  CD 0000*      CALL  outbyte
0C58'  C1            POP   BC
0C59'  2A 0C62'      LD    HL,(Optr)
0C5C'  23            INC   HL
0C5D'  22 0C62'      LD    (Optr),HL
0C60'  18 E9         JR    CC192
0C62'              Optr:   ds      2
0C64'  C1          outhex::POP     BC
0C65'  E1            POP   HL
0C66'  E5            PUSH  HL
0C67'  C5            PUSH  BC      ;Return Stack
0C68'  24            INC   H
0C69'  25            DEC   H
0C6A'  28 19         JR    Z,_HEX2
0C6C'  7C            LD    A,H
0C6D'  FE A0         CP    0A0H
0C6F'  E5            PUSH  HL
0C70'  11 0030       LD    DE,'0'
0C73'  D5            PUSH  DE
0C74'  D4 0000*      CALL  NC,outbyte
0C77'  D1            POP   DE
0C78'  E1            POP   HL
0C79'  7C            LD    A,H
0C7A'  E5            PUSH  HL
0C7B'  CD 0C9F'      CALL  OUTHEX2
0C7E'  E1            POP   HL
0C7F'  7D            LD    A,L
0C80'  CD 0C9F'      CALL  OUTHEX2
0C83'  18 11         JR    _RET
0C85'  7D          _HEX2:  LD      A,L
0C86'  FE A0         CP    0A0H
0C88'  E5            PUSH  HL
0C89'  11 0030       LD    DE,'0'
0C8C'  D5            PUSH  DE
0C8D'  D4 0000*      CALL  NC,outbyte
0C90'  D1            POP   DE
0C91'  E1            POP   HL
0C92'  7D            LD    A,L
0C93'  CD 0C9F'      CALL  OUTHEX2
0C96'  21 0048     _RET:   LD      HL,'H'
0C99'  E5            PUSH  HL
0C9A'  CD 0000*      CALL  outbyte
0C9D'  C1            POP   BC
0C9E'  C9            RET
0C9F'  F5          OUTHEX2:PUSH    AF
0CA0'  0F            RRCA
0CA1'  0F            RRCA
0CA2'  0F            RRCA
0CA3'  0F            RRCA
0CA4'  CD 0CA8'      CALL  HEX21
0CA7'  F1            POP   AF
0CA8'  E6 0F       HEX21:  AND     0FH
0CAA'  C6 30         ADD   A,'0'
0CAC'  FE 3A         CP    3AH
0CAE'  38 02         JR    C,$+4
0CB0'  C6 07         ADD   A,7
0CB2'  26 00         LD    H,0
0CB4'  6F            LD    L,A
0CB5'  E5            PUSH  HL
0CB6'  CD 0000*      CALL  outbyte
0CB9'  E1            POP   HL
0CBA'  C9            RET
                     EXT   clearsta
                     EXT   findglb
                     EXT   search
                     EXT   poll
                     EXT   const
                     EXT   getlabel
                     EXT   col
                     EXT   nl
                     EXT   outbyte
                     EXT   cout
                     EXT   outdec
                     END
```

▶付録ディスクはとてもよいです。今後もどんどん付けてください。なかでもSX-WINDOW用アクセサリがよく，今後もSX-WINDOW上のゲームやウインドウの機能をいろいろ細かく拡張できるツールを付けてほしいと思います。　　伊藤　雅裕(29)東京都

リスト7 RDRTL.MAC

```
        ;RDRTL.ASM  Small C Runtime Library with redirectable output
        ;       Always used with compiler
        ;       Must be linked first when used
        ;       Supports 8 256 Bytes file buffers
        ;       Redirected output via > (only to a file)
        ;       Redirected append output via >> (only to a file)
        ;       Functions included:
        ;          fopen()         fclose()        (r, w, & a modes)
        ;          getchar()       putchar()
        ;          gets()          puts()
        ;          getc()          putc()
        ;          fgetc()         fputc()
        ;          fgets()         fputs()
        ;          exit()          abort()
        ;          and all supporting functions
        ;          grabio(), freeio()
        ;                  by
        ;          F. A. Scacchitti
        ;          25 Glenview Lane
        ;          Rochester, NY 14609
        ;             3 - 25 - 86
        ;    ***
             EXT     zzbuf
        ;    EXT     CALL            ;WZD C^ 1>]L'Y=YD7J :I7^.3&B9Y
0000  RDRTL::                                ;RDRTL 6^ DX:OZ0:D&<R=
                ;    ***
                ; ========================================
                ;  I/O subroutines for CP/M
                ;   originally
                ;           By Glen Fisher
                ;           The Code Works(tm)
                ;   then
                ;           [modified by Bill Randle]
                ;   and finally
                ;   [significantly modified by Fred Scacchitti]
                ;   ported on S-OS "Sword" System
                ;           by T.Ishigami 2 - 20 - 91
                ; ========================================
0000            NULL    EQU 0           ;pointer to nothing
0038            InfSize EQU 38H         ;size, in bytes, of Information Block
0039            NEXTP   EQU InfSize+1 ;offset to next-character pointer in I/O stru
cture
003B            UNUSED  EQU InfSize+3 ;offset to unused-positions-count in I/O stru
cture
003C            ERRFLG  EQU InfSize+4 ;offset to Error-flg in I/O structure
003D            UNGOT   EQU InfSize+5 ;offset to char ungotten by ungetc()
003E            FLAG    EQU InfSize+6 ;file-type flag       byte (in unused part of
 FCB)
003F            FCBSIZE EQU InfSize+7 ;size, in bytes, of an FCB
0080            FREEFLG EQU 128         ;This I/O structure is available for the taki
ng
0002            EOFFLG  EQU 2           ;The end of this file has been hit
0001            WRTFLG  EQU 1           ;This file open       for writing
0100            BUFSIZ  EQU 256         ;how long the sector buffer is
0008            NBUFS   EQU 8           ;number       of I/O buffers
0008            NSECTS  EQU 8           ;sectors per buffer
0100            TBUFSZ  EQU 256         ;default S-OS buffer size
                                        ; (change buffer declarations, too)
1FFA            _HOT    EQU 1FFAH
1FF4            _PRINT  EQU 1FF4H
1FEE            _LTNL   EQU 1FEEH
1FE5            _MSX    EQU 1FE5H
1FDC            _LPRNT  EQU 1FDCH
1FD3            _GETL   EQU 1FD3H
2021            _FLGET  EQU 2021H
1FA3            _FILE   EQU 1FA3H
1FAF            _WOPEN  EQU 1FAFH
2000            _DRDSB  EQU 2000H
2003            _DWTSB  EQU 2003H
2015            _KILL   EQU 2015H
1F76            _KBFAD  EQU 1F76H
1F74            _IBFAD  EQU 1F74H
1F64            _DTBUF  EQU 1F64H
1F62            _FATBF  EQU 1F62H
1F6A            _MEMAX  EQU 1F6AH
1F60            _DIRPS  EQU 1F60H
1F5E            _FATPOS EQU 1F5EH
1F5D            _DSK    EQU 1F5DH
                ;
000D            EOL     EQU 13  ;end-of-line character (=carriage return)
001A            CTRLZ   EQU 26      ;end-of-file mark for text files
FFFF            EOF     EQU -1  ;end of file marker for other
0018            MAXARG  EQU 24      ;Maximum number of args in input line
0001            STDOUT  EQU  1      ;Default for stdout
0002            STDERR  EQU  2      ;Default for stderr
                ;
                ; First     thing, we save SWORD's stack pointer and
                ; current disk and init stdin and stdout.
                ; Second thing, we run through the SWORD input line and
                ; modify it so that we can pass     the C program the
                ; command line in the argc, argv form that it expects
                ; HL = pointer to next argv entry
                ; DE = pointer to next character in command line
                ; B = number of     characters left in line
                ; C = argument count (argc)
0004 73 002D'         ULINK:: LD      (zzstak),SP                 : Save
stack pointer for later
   ;
0008   SETIO:
0008                  LD      B,NBUFS
013E*                 LD      HL,zzbuf+TBUFSZ+FLAG
013F                  LD      DE,FCBSIZE+BUFSIZ
000C' 3E 80           LD      A,FREEFLG
000E' 77      SETIO2: LD      (HL),A          :# SET ALL BUFFERS TO FREE
000F' 19              ADD     HL,DE           :on to next buffer
0010' 10 FC           DJNZ    SETIO2          :if there is one...
                ;
0012' 23              INC     HL
0013' 22 002B'        LD      (zzmem),HL
                ;
0016' 3A 1F5D         LD      A,(_DSK)        ; get logged-in disk
0019' 32 0817'        LD      (DFLTDSK),A
                ;
001C' ED 7B 1F6A      LD      SP,(_MEMAX)     ; Set stack pointer
                ;
0020' 21 0001         LD      HL,STDOUT
0023' 22 0829'        LD      (RSTDOUT),HL    ; Init rstdout
                ;
0026' 0E 00           LD      C,0             ; Init argc
0028' 21 0068'        LD      HL,ARGV         ; Pointer to first entry of arg
v array
                ; Ok, now for the real stuff.  Set DE pair to point to
                ; SWORD command line and start searching for arguments
002B' ED 5B 1F76      LD      DE,(_KBFAD)     ; Pointer to SWORD arg line
002F' 13              INC     DE
0030' 13              INC     DE
0031' 1A      NXTSP:  LD      A,(DE)          ; Load # character in line
0032' 13              INC     DE
0033' A7              AND     A
0034' 28 1B           JR      Z,ENDCMD        ; End of cmd line
0036' FE 20           CP      ' '             ; Space?
0038' 28 F7           JR      Z,NXTSP         ; Yes...continue searching
003A' FE 3E           CP      '>'             ; Redirect output?
003C' CA 0098'        JP      Z,RDOUT         ; Yes...open the file for outpu
t
003F' CD 0060'        CALL    SVARG           ; Nope, save starting point of
this arg
                  ; Loop looking for either end of line of a space
0042' 1A          NXTCH:      LD      A,(DE)  ; Load next character in line
0043' 13              INC     DE              ; Point to next character
0044' A7              AND     A
0045' 28 0A           JR      Z,ENDCMD        ; End of cmd line, but need to
end arg
0047' FE 20           CP      ' '             ; Space?
0049' 20 F7           JR      NZ,NXTCH        ; No...keep looking
004B' 1B              DEC     DE

004C' AF              XOR     A               ; Yes, replace it with a zero b
yte
004D' 12              LD      (DE),A

004E' 13              INC     DE
004F' 18 E0           JR      NXTSP           ; Look for start of next arg

0051' 06 00       ENDCMD:     LD      B,0     ; Zero B (BC now is 16 bit argc
)
0053' C5              PUSH    BC              ; First arg to main procedure
0054' 21 0068'        LD      HL,ARGV         ; Point to argv array
0057' E5              PUSH    HL              ; Second argument to main proce
dure
0058' 3E 02           LD      A,2             ; Load up the argument count
005A' CD 0000*        CALL    main##          ; Transfer to the C world....
                  ;   off we go into the wild  b l u e
005D' C3 0137'        JP      exit            ; Return to CPM
                  ;
0060' 1B          SVARG:
0060' 1B              DEC     DE
0061' 73              LD      (HL),E          ; Save pointer to start of stri
ng
0062' 23              INC     HL
0063' 72              LD      (HL),D
0064' 23              INC     HL
0065' 13              INC     DE
0066' 0C              INC     C               ; Increment argc
0067' C9              RET
0068'             ARGV:
0068'                 DS      MAXARG*2
                  ;
0098' 1A          RDOUT:      LD      A,(DE)
0099' FE 3E           CP      '>'             ; check for append mode
009B' C2 00A4'        JP      NZ,RDOUT1
009E' 3E 0F           LD      A,0FH
00A0' 32 0826'        LD      (AFLAG),A       ; Set flag for appending
00A3' 13              INC     DE              ; stand on next character

00A4' C5          RDOUT1:     PUSH    BC
00A5' E5              PUSH    HL
00A6' CD 00D6'        CALL    DEBLNK          ; Skip leading spaces
00A9' CD 00E1'        CALL    CPYNAM          ; Copy filename to temp buffer
00AC' D5              PUSH    DE              ; Save registers
00AD' 21 0104'        LD      HL,NAMBUF       ; Begining of filename
00B0' E5              PUSH    HL
00B1' 3A 0826'        LD      A,(AFLAG)       ; Test for append mode
00B4' FE 0F           CP      0FH
00B6' 21 0100'        LD      HL,WROPN
00B9' 20 07           JR      NZ,RDOUT3       ; No, set write mode
00BB' AF              XOR     A               ; Yes, clear the flag and
00BC' 32 0826'        LD      (AFLAG),A
00BF' 21 0102'        LD      HL,APOPN        ; set append mode

00C2' E5          RDOUT3:     PUSH    HL
00C3' CD 0198'        CALL    fopen           ; Open the file
00C6' C1              POP     BC
00C7' C1              POP     BC
00C8' 7C              LD      A,H
00C9' B5              OR      L               ; Check return status
00CA' CA 00F3'        JP      Z,RDERR
00CD' 22 0829'        LD      (RSTDOUT),HL    ; Save unit for redirected outp
ut
00D0' D1              POP     DE              ; Restore registers
00D1' E1              POP     HL
00D2' C1              POP     BC
00D3' C3 0031'        JP      NXTSP           ; Get next argument
                  ;
00D6'             DEBLNK:
00D6' 1A              LD      A,(DE)
00D7' A7              AND     A
00D8' CA 00F3'        JP      Z,RDERR         ; End of command line ?
00DB' FE 20           CP      ' '
00DD' C0              RET     NZ
00DE' 13              INC     DE              ; Skip leading spaces
00DF' 18 F5           JR      DEBLNK
                  ;
00E1' 21 0104'    CPYNAM:     LD      HL,NAMBUF       ; Copy  filename to tem
p buffer
00E4' 06 10           LD      B,16            ; Maximum filename length
00E6' 1A          CPY1:       LD      A,(DE)
00E7' 36 00           LD      (HL),0
00E9' FE 20           CP      ' '
00EB' C8              RET     Z
00EC' A7              AND     A
00ED' C8              RET     Z
00EE' 77              LD      (HL),A
00EF' 13              INC     DE
00F0' 23              INC     HL
00F1' 10 F3           DJNZ    CPY1
                  ;
00F3' 11 0114'    RDERR:      LD      DE,RDEMSG       ; Error message
00F6' CD 1FE5         CALL    _MSX            ; Make sure it gets put on the
terminal (mod fas)
00F9' C3 0137'        JP      exit
                  ;
00FC' 00 00       TEMP:       DB      0,0
00FE' 52 00       RDOPN:      DB      'R',0
0100' 57 00       WROPN:      DB      'W',0
0102' 41 00       APOPN:      DB      'A',0
0104'             NAMBUF:     DS      16
0114' 52 44 52 54 RDEMSG:     DB      'RDRTL: UNABLE TO OPEN OUTPUT FILE',0DH
,0
0118' 4C 3A 09 55
011C' 4E 41 42 4C
0120' 45 20 54 4F
0124' 20 4F 50 45
0128' 4E 20 4F 55
012C' 54 50 55 54
0130' 20 46 49 4C
0134' 45 0D 00
                  ;    exit()
                  ;    Stop execution of the program,
                  ;    restore the logged-in drive,
                  ;    and hot-start SWORD
0137' 2A 0829'    exit::      LD      HL,(RSTDOUT)
013A' 7C              LD      A,H
013B' B7              OR      A                       ; See if stdout has bee
n redirected
013C' 28 05           JR      Z,exit1
013E' E5              PUSH    HL
013F' CD 0262'        CALL    fclose                  ; If so, close the file
0142' C1              POP     BC
0143' 3A 0817'    exit1:      LD      A,(DFLTDSK)             ; Grab orig. lo
gged-in disk
0146' 32 1F5D         LD      (_DSK),A                ; and log it in again
0149' ED 7B 002D'     LD      SP,(zzstak)             ; Load stack pointer
014D' C3 1FFA         JP      _HOT                    ; return to SWORD

                  ;    abort(reason)
0150' C1          abort::     POP     BC
0151' E1              POP     HL
0152' 7C              LD      A,H
0153' B5              OR      L
0154' 28 04           JR      Z,abort2
0156' 7D              LD      A,L
0157' 32 016C'        LD      (ABCODE),A
015A' 11 0163'    abort2:     LD      DE,ABTMSG               ; Load abort me
ssage
015D' CD 1FE5         CALL    _MSX
0160' C3 0137'        JP      exit

0163' 0D 41 42 4F ABTMSG:     DB      0DH, 'ABORTED '
0167' 52 54 45 44
016B' 20
016C' 07 0D 00    ABCODE:     DB      07H,0DH,0
                  ;
                  ;    grabio()
                  ;    find an input buffer, and return its address.
                  ;    if there isn't one, return a NULL.
016F'             grabio::                            ;6 May 80 rj
016F' 06 08           LD      B,NBUFS
0171' 21 013E*        LD      HL,zzbuf+TBUFSZ+FLAG
0174' 11 013F         LD      DE,FCBSIZE+BUFSIZ
0177' 3E 80           LD      A,FREEFLG
0179' BE          GRAB2:      CP      (HL)            ;flag byte == freeflg?
017A' 28 07           JR      Z,GRAB3                 ;if so, found a free bu
ffer
017C' 19              ADD     HL,DE                   ;on to next buffer
017D' 10 FA           DJNZ    GRAB2                   ;if there is one...
017F' 21 0000         LD      HL,NULL                 ;there ain't
0182' C9              RET                             ;give up
                  ;
0183' 36 00       GRAB3:      LD      (HL),0                  ;mark buffer as
 taken
0185' 11 FFC2         LD      DE,-FLAG                ;back up to buffer star
t
0188' 19              ADD     HL,DE
0189' C9              RET                             ;and hand it back
                  ;
                  ;    freeio(unit)
                  ;    mark a buffer as free.
018A'             freeio::                            ;Mod  6 May 80 rj
018A' C1              POP     BC                      ;save rtn addr
018B' E1              POP     HL                      ;get buffer addr
018C' E5              PUSH    HL                      ;put the stack back tog
ether
018D' C5              PUSH    BC
018E' 11 003E         LD      DE,FLAG                 ;find flag byte
0191' 19              ADD     HL,DE
0192' 36 80           LD      (HL),FREEFLG            ;mark buffer as 'free'
0194' 21 0000         LD      HL,NULL                 ;return something
0197' C9              RET
                  ;
                  ;    fopen(name,mode)
0198'             fopen::
0198' C1              POP     BC                      ;get args
0199' E1              POP     HL                      ;mode
019A' D1              POP     DE
019B' 22 081E'        LD      (MODE),HL
019E' ED 53 081C'     LD      (FILE),DE
01A2' E5              PUSH    HL
01A3' D5              PUSH    DE
01A4' C5              PUSH    BC
01A5' CD 016F'        CALL    grabio                  ; unit = grabio();
01A8' 22 0818'        LD      (unit),HL
01AB' 7C              LD      A,H                     ; if(unit==NULL)
01AC' B5              OR      L                       ;       return(NULL);
01AD' C8              RET     Z
01AE' E5              PUSH    HL
01AF' DD E1           POP     IX
01B1' DD 36 3D 00     LD      (IX+UNGOT),0            ;# EOF TO AVOID EXTRA C
HAR
01B5' DD 36 3C 00     LD      (IX+ERRFLG),0           ;# Clear ERROR BYTE
01B9' 11 003F         LD      DE,FCBSIZE
01BC' 19              ADD     HL,DE
01BD' DD 75 39        LD      (IX+NEXTP),L
01C0' DD 74 3A        LD      (IX+NEXTP+1),H
01C3' 2A 081E'        LD      HL,(MODE)               ;if(*mode=='R'||*mode==
'A'){
01C6' 7E              LD      A,(HL)
01C7' E6 5F           AND     5FH                     ;# CONVERT TO UPPERCASE
01C9' FE 52           CP      'R'                     ;# MODE = R ?
01CB' C2 01E3'        JP      NZ,FOPIF1
                  ;
01CE'             FOPIF0:
01CE' 3E 04           LD      A,4
01D0' ED 5B 081C'     LD      DE,(FILE)               ;       if(cpm(OPEN,uni
t)<0){
01D4' 2A 0818'        LD      HL,(unit)
01D7' CD 0438'        CALL    _RDOPEN
01DA' 38 7E           JR      C,FOPIF5
01DC' DD 36 3B 00     LD      (IX+UNUSED),0
                                                      ;       }
01E0' C3 0256'        JP      FOPIFX
01E3'             FOPIF1:                                     ; else if(*mode
=='W'){
01E3' E6 5F           AND     5FH
01E5' FE 57           CP      'W'                     ;# WRITE MODE ?
01E7' CA 0203'        JP      Z,FOPIFA                ;# YES - GO DO IT
01EA' FE 41           CP      'A'                     ;# APPEND MODE ?
01EC' C2 025A'        JP      NZ,FOPIF5               ;# NO  - BACK TO CALLER
 OF fopen
                  ;                                   ;#     NO MODES LEFT TO
 TRY
01EF' 3E 04           LD      A,4
01F1' ED 5B 081C'     LD      DE,(FILE)
01F5' 2A 0818'        LD      HL,(unit)               ;# FIRST LET'S SEE IF I
T'S THERE
01F8' CD 0438'        CALL    _RDOPEN
01FB' D2 0211'        JP      NC,FOPIF3               ;# YES  - SET IT UP FOR
 USE
                                                      ;# NO   - LET'S MAKE ON
E
01FE' 2A 081E'        LD      HL,(MODE)               ;# SET MODE TO 'W' ON N
EW FILE
0201' 36 57           LD      (HL),'W'                ;# TO AVOID WASTING TIM
E AND CODE
                                                      ;# SEARCHING AN EMPTY F
ILE
                  ;
0203'             FOPIFA:
0203' 3E 04           LD      A,4
0205' ED 5B 081C'     LD      DE,(FILE)
0209' 2A 0818'        LD      HL,(unit)
020C' CD 0521'        CALL    _WROPEN
020F' 38 49           JR      C,FOPIF5
0211'             FOPIF3:
0211' DD 36 3B 00     LD      (IX+UNUSED),0
0215' DD 7E 3E        LD      A,(IX+FLAG)             ;unit[FLAG] = WRITE_FL;
0218' F6 01           OR      WRTFLG
021A' DD 77 3E        LD      (IX+FLAG),A
021D' 11 003F         LD      DE,FCBSIZE
0220' 2A 0818'        LD      HL,(unit)
0223' 19              ADD     HL,DE
0224' DD 75 39        LD      (IX+NEXTP),L
0227' DD 74 3A        LD      (IX+NEXTP+1),H          ;unit[NEXTP] = &unit +
FCBSIZE;
022A' 2A 081E'        LD      HL,(MODE)               ;#
022D' 7E              LD      A,(HL)                  ;# GET MODE
022E' E6 5F           AND     5FH
0230' FE 41           CP      'A'                     ;# APPEND MODE ?
0232' 20 22           JR      NZ,FOPIFX               ;# NO  -  RETURN NORMAL
LY
                  ;
0234' DD 7E 36        LD      A,(IX+36H)              ;RC
0237' DD 77 32        LD      (IX+32H),A              ;FLPNT
023A' CD 05FE'        CALL    _READ
023D' 38 1B           JR      C,FOPIF5
023F' DD 34 32        INC     (IX+32H)                ;FLPNT++;
0242' 2A 0818'        LD      HL,(unit)
0245' 11 003F         LD      DE,FCBSIZE
0248' 19              ADD     HL,DE
0249' ED 5B 07F1'     LD      DE,(FLSIZE)
024D' 16 00           LD      D,0
024F' 19              ADD     HL,DE
0250' DD 75 39        LD      (IX+NEXTP),L
0253' DD 74 3A        LD      (IX+NEXTP+1),H
                  ;
0256'             FOPIFX:
0256' 2A 0818'        LD      HL,(unit)               ; return(unit);
0259' C9              RET

025A'             FOPIF5:
025A' DD 36 3E 80     LD      (IX+FLAG),FREEFLG       ; freeio(unit);
025E' 21 0000         LD      HL,NULL                 ; return(NULL);
0261' C9              RET
                  ;
                  ;    fclose(unit)
0262'             fclose::
0262' C1              POP     BC
0263' E1              POP     HL
0264' 22 0818'        LD      (unit),HL
0267' E5              PUSH    HL
0268' C5              PUSH    BC
0269' 7C              LD      A,H                     ; if (unit<256)
026A' B7              OR      A                       ; /* assume stdin, stdo
ut, etc. */
026B' 21 0000         LD      HL,0
026E' C8              RET     Z                       ;       return NULL;
026F' DD 2A 0818'     LD      IX,(unit)               ; if(unit[FLAG] & WRITE
_FL){
0273' DD 7E 3E        LD      A,(IX+FLAG)
0276' E6 01           AND     WRTFLG
0278' 28 2A           JR      Z,FCLIF1
027A' 21 0000         LD      HL,0
027D' E5              PUSH    HL                      ;       putc(EOF,unit);

027E' DD E5           PUSH    IX
0280' CD 03A3'        CALL    cput
0283' DD E1           POP     IX
0285' C1              POP     BC
0286' DD 7E 3B        LD      A,(IX+UNUSED)
0289' DD 77 12        LD      (IX+12H),A              ;       Low Byte of the
 File Size
028C' A7              AND     A
028D' 28 0C           JR      Z,FCLIF3
028F' ED 5B 0818'     LD      DE,(unit)
0293' CD 0665'        CALL    _WRITE
0296' 38 14           JR      C,FCLIF2
0298' DD 34 32        INC     (IX+32H)                ;       FLPNT++;

029B'             FCLIF3:
029B' ED 5B 0818'     LD      DE,(unit)
029F' CD 059A'        CALL    _CLOSE                  ; if(close(unit) < 0)
02A2' 38 08           JR      C,FCLIF2
02A4'             FCLIF1:
02A4' DD 36 3E 80     LD      (IX+FLAG),FREEFLG       ; freeio(unit);
02A8' 21 0001         LD      HL,1                    ; return(YES);
02AB' C9              RET
                  ;
02AC'             FCLIF2:
02AC' DD 36 3E 80     LD      (IX+FLAG),FREEFLG       ; freeio(unit);
02B0' 21 0000         LD      HL,0                    ; return(NULL);
02B3' C9              RET
                  ;
                  ;    getc(unit)
02B4'             fgetc::
02B4'             getc::
02B4' C1              POP     BC
02B5' E1              POP     HL
02B6' E5              PUSH    HL
02B7' C5              PUSH    BC
02B8' 7C              LD      A,H                     ; file or console ?
02B9' B7              OR      A
02BA' CA 0313'        JP      Z,getchar               ; console

02BD' E5          CGET1:      PUSH    HL
02BE' DD E1           POP     IX
02C0' DD 7E 3E        LD      A,(IX+FLAG)
02C3' E6 02           AND     EOFFLG                  ;if(unit[FLAG]  & EOF_F
L)
02C5' 21 FFFF         LD      HL,-1                   ;       return(-1);
02C8' C0              RET     NZ
```

▶PC-9801用マウスがつながるということは，PC-9801用トラックボールがつなげるということではないか！　ラッキー！
白川　雄資(18)愛知県

```
                        ;
02C9' DD 7E 3D              LD      A,(IX+UNGOT)        ; check for ungotten ch
ar
02CC' A7                    AND     A                   ; is it an end of file
?
02CD' 28 08                 JR      Z,GTCCON
02CF' DD 36 3D 00           LD      (IX+UNGOT),0
02D3' 6F                    LD      L,A
02D4' 26 00                 LD      H,0
02D6' C9                    RET                         ; return with char in H
L
                        ;
02D7' DD 6E 39          GTCCON:     LD      L,(IX+NEXTP)
02DA' DD 66 3A              LD      H,(IX+NEXTP+1)
02DD' DD 7E 3B              LD      A,(IX+UNUSED)
02E0' A7                    AND     A
02E1' 20 14                 JR      NZ,GTCIF2
02E3' DD E5                 PUSH    IX
02E5' D1                    POP     DE
02E6' CD 05FE'              CALL    _READ
02E9' 21 FFFF               LD      HL,-1               ;               return(
-1);
02EC' D8                    RET     C
02ED' DD 34 32              INC     (IX+32H)            ;               FLPNT++
;
02F0' DD E5                 PUSH    IX
02F2' E1                    POP     HL
02F3' 11 003F               LD      DE,FCBSIZE
02F6' 19                    ADD     HL,DE
                                                        ;       }
02F7'                   GTCIF2:
02F7' DD 35 3B              DEC     (IX+UNUSED)         ; unit[UNUSED]--;
02FA' 7E                    LD      A,(HL)
02FB' 23                    INC     HL
02FC' DD 75 39              LD      (IX+NEXTP),L
02FF' DD 74 3A              LD      (IX+NEXTP+1),H
0302' A7                    AND     A
0303' 6F                    LD      L,A                 ; return(*cp & 0xff);
0304' 26 00                 LD      H,0
0306' C0                    RET     NZ
0307' DD 7E 3E              LD      A,(IX+FLAG)         ;       unit[FLAG] |= E
OF_FL;
030A' F6 02                 OR      EOFFLG
030C' DD 77 3E              LD      (IX+FLAG),A
030F' 21 FFFF               LD      HL,-1               ;       return(-1);
0312' C9                    RET
                                                        ;       }
                        ;
                        ;   getchar()
0313'                   getchar::                       ;  read from console
0313' CD 2021               CALL    _FLGET              ;  getcH() & 0xff;
0316' CD 1FF4               CALL    _PRINT
0319' 6F                    LD      L,A
031A' 26 00                 LD      H,0
031C' FE 1A                 CP      CTRLZ               ; if(t==CTRLZ)
031E' C0                    RET     NZ
031F' 21 FFFF               LD      HL,-1               ;       return(EOF);
0322' C9                    RET
                        ;
                        ;   gets(buff)
0323'                   gets::
0323' C1                    POP     BC
0324' D1                    POP     DE
0325' D5                    PUSH    DE
0326' C5                    PUSH    BC
0327' CD 1FD3               CALL    _GETL
032A' 1A                    LD      A,(DE)
032B' FE 1B                 CP      1BH
032D' CA 0137'              JP      Z,exit
0330' CD 1FEE               CALL    _LTNL               ;  putchar('¥n')
0333' 62                    LD      H,D                 ; return(buff);
0334' 6B                    LD      L,E
0335' C9                    RET                         ; }
                        ;
                        ;   fgets(cp,len,unit)
0336'                   fgets::
0336' D1                    POP     DE                  ; skip rtn addr
0337' DD E1                 POP     IX                  ; unit
0339' C1                    POP     BC                  ; length
033A' E1                    POP     HL                  ; cp
033B' E5                    PUSH    HL
033C' C5                    PUSH    BC
033D' DD E5                 PUSH    IX
033F' D5                    PUSH    DE
0340' 22 0827'              LD      (SVCHP),HL          ;       save_cp = cp;

0343' 0B                fgets2:     DEC     BC                  ;       while (
--len) {
0344' 78                    LD      A,B
0345' B1                    OR      C
0346' CA 0371'              JP      Z,fgets4
0349' C5                    PUSH    BC
034A' E5                    PUSH    HL                  ; save cp
034B' DD E5                 PUSH    IX                  ; unit
034D' CD 02B4'              CALL    getc                ;           c = getc(un
it);
0350' 7C                    LD      A,H                 ;           if(c==EOF)
/* c>255 */
0351' B7                    OR      A
0352' 7D                    LD      A,L
0353' DD E1                 POP     IX
0355' E1                    POP     HL
0356' C1                    POP     BC
0357' CA 036A'              JP      Z,fgets3
035A' ED 5B 0827'           LD      DE,(SVCHP)          ;               if (c
p<>save_cp)
035E' E5                    PUSH    HL
035F' B7                    OR      A
0360' ED 52                 SBC     HL,DE
0362' E1                    POP     HL
0363' C2 0371'              JP      NZ,fgets4           ;               else
0366' 21 0000               LD      HL,0                ;
 /* no characters */
0369' C9                    RET                         ;
 return (NULL);
036A' 77                fgets3:     LD      (HL),A              ;
 *cp++ = c;
036B' 23                    INC     HL
036C' FE 0D                 CP      EOL                 ;               if(c=='
¥n')
036E' C2 0343'              JP      NZ,fgets2
0371' 36 00             fgets4:     LD      (HL),0              ;
        *cp='¥0';
0373' 2A 0827'              LD      HL,(SVCHP)          ;
 return save_cp;
0376' C9                    RET                         ; }     }       }
 }
                        ;
                        ;   putc(c,unit)
0377'                   fputc::
0377'                   putc::
0377' C1                    POP     BC                  ;rtn addr
0378' D1                    POP     DE                  ;unit
0379' E1                    POP     HL                  ;c
037A' E5                    PUSH    HL
037B' D5                    PUSH    DE
037C' C5                    PUSH    BC
037D' 7A                    LD      A,D
037E' B7                    OR      A                   ; if(unit < 256) {
037F' 20 14                 JR      NZ,putc4            ;       /* assume stdou
t, stderr */
0381' 7B                    LD      A,E
0382' FE 01                 CP      STDOUT              ;       if(unit == stdo
ut) {
0384' 20 06                 JR      NZ,putc1
0386' E5                    PUSH    HL
0387' CD 03E8'              CALL    putchar             ;               putchar
(c);
038A' E1                    POP     HL
038B' C9                    RET                         ;               return;
}

038C' FE 02             putc1:      CP      STDERR              ;       elseif(
unit == stderr) {
038E' 7D                    LD      A,L
038F' CA 1FF4               JP      Z,_PRINT                    ;
 putconsole(c);
0392' C3 039F'              JP      putcerr             ;       else goto putce
rr; }

0395' E5                putc4:      PUSH    HL                  ; if(cput(c,uni
t)<0)
0396' D5                    PUSH    DE                  ;       goto putcerr;
0397' CD 03A3'              CALL    cput
039A' D1                    POP     DE
039B' 7C                    LD      A,H
039C' B7                    OR      A
039D' E1                    POP     HL                  ; return(c);
039E' C8                    RET     Z

039F'                   putcerr:
039F' 21 FFFF               LD      HL,-1               ;putcerr:
03A2' C9                    RET                         ; return(-1);
                        ;
                        ;   cput(c,unit)
03A3'                   cput::
03A3' C1                    POP     BC
03A4' DD E1                 POP     IX
03A6' E1                    POP     HL
03A7' E5                    PUSH    HL
03A8' DD E5                 PUSH    IX
03AA' C5                    PUSH    BC
03AB' 7D                    LD      A,L
03AC' 32 0820'              LD      (ZCH),A
03AF' DD 22 0818'           LD      (unit),IX
03B3' DD 6E 39              LD      L,(IX+NEXTP)
03B6' DD 66 3A              LD      H,(IX+NEXTP+1)
03B9' 77                    LD      (HL),A              ; return((*cp = c) & 0x
ff);
03BA' 23                    INC     HL
03BB' DD 75 39              LD      (IX+NEXTP),L
03BE' DD 74 3A              LD      (IX+NEXTP+1),H
03C1' DD 34 3B              INC     (IX+UNUSED)         ; if(unit[UNUSED]++ ==
0){
03C4' 20 1E                 JR      NZ,PTCIF1
03C6' ED 5B 0818'           LD      DE,(unit)
03CA' CD 0665'              CALL    _WRITE
03CD' 21 FFFF               LD      HL,-1               ;               return(
-1);
03D0' D8                    RET     C

03D1' DD 34 32              INC     (IX+32H)            ;               FLPNT++
;
03D4' DD E5                 PUSH    IX
03D6' E1                    POP     HL
03D7' 11 003F               LD      DE,FCBSIZE
03DA' 19                    ADD     HL,DE
03DB' DD 75 39              LD      (IX+NEXTP),L
03DE' DD 74 3A              LD      (IX+NEXTP+1),H
03E1' 3A 0820'              LD      A,(ZCH)             ;       POP zch
03E4'                   PTCIF1:
03E4' 26 00                 LD      H,0
03E6' 6F                    LD      L,A
03E7' C9                    RET
                        ;
                        ; putchar(c)
03E8'                   putchar::
03E8' C1                 POP BC
03E9' E1                 POP HL
03EA' E5                 PUSH HL
03EB' C5                 PUSH BC
03EC' 3A 082A'           LD A,(RSTDOUT+1)  ; if(rdstdout >= 256) {
03EF' A7                 AND A   ; /* stdout has been redirected */
03F0' 28 0C              JR Z,putcon

03F2' E5                 PUSH HL
03F3' ED 5B 0829'        LD DE,(RSTDOUT)
03F7' D5                 PUSH DE
03F8' CD 0377'           CALL putc   ; putc(c. rstdout);
03FB' C1                 POP BC
03FC' C1                 POP BC   ; return;
03FD' C9                 RET
                            ; } else {
03FE' 7D                putcon: LD A,L   ; print(PUTCH,c);
03FF' CD 1FF4            CALL _PRINT   ; (mod fas)
0402' 26 00              LD H,0   ; return(c & 0xff);
0404' C9                 RET    ; }
                        ;
                        ; puts(cp)
0405'                   puts::
0405' C1                 POP BC   ; get args
0406' E1                 POP HL
0407' E5                 PUSH HL
0408' C5                 PUSH BC
0409' E5                 PUSH HL   ; cp
040A' 2A 0829'           LD HL,(RSTDOUT)
040D' E5                 PUSH HL
040E' CD 0414'           CALL fputs   ; return (fputs(cp, rstdout));
0411' C1                 POP BC
0412' C1                 POP BC
0413' C9                 RET
                        ;
                        ; fputs(cp,unit)
0414'                   fputs::
0414' C1                 POP BC
0415' D1                 POP DE   ; unit
0416' E1                 POP HL   ; cp
0417' E5                 PUSH HL
0418' D5                 PUSH DE
0419' C5                 PUSH BC

041A' 7E                fputs1: LD A,(HL)   ; while((c=*cp++) <> NULL) {
041B' 23                 INC HL
041C' B7                 OR A
041D' CA 0434'           JP Z,fputs3
0420' E5                 PUSH HL
0421' 4F                 LD C,A
0422' 06 00              LD B,0
0424' C5                 PUSH BC
0425' D5                 PUSH DE
0426' CD 0377'           CALL putc   ; if(putc(c,unit)==EOF)
0429' 7C                 LD A,H
042A' B7                 OR A
042B' D1                 POP DE
042C' C1                 POP BC
042D' E1                 POP HL
042E' 28 EA              JR Z,fputs1
0430' 21 FFFF            LD HL,-1
0433' C9                 RET    ;  return(EOF);

0434' 21 0000           fputs3: LD HL,0
0437' C9                 RET    ; return(NULL);
                        ;**********************************
                        ; S-OS 'Sword' Simulation Program
                        ; Programed by T.Ishigami
                        ;   '90 Mar 2nd
                        ;**********************************
                        ;===================
                        ; FILE OPEN FOR READ
                        ;===================
0438'                   _RDOPEN:
0438' 22 07DC'           LD (FCB_ADR),HL
043B' CD 1FA3            CALL _FILE
043E' D8                 RET C
043F' CD 04A0'           CALL ROPEN
0442' D8                 RET C
                         ;
0443' 2A 1F74            LD HL,(_IBFAD)
0446' 11 07DF'           LD DE,FILE_BF
0449' 01 0020            LD BC,20H
044C' ED B0              LDIR
044E' 3A 1F5D            LD A,(_DSK)
0451' 32 0810'           LD (FLDSK),A
0454' AF                 XOR A
0455' 32 07DE'           LD (LST_DSK),A ;NEVER BEING THE SAME
0458' CD 0777'           CALL RDFAT
045B' D8                 RET C

045C' 06 10              LD B,10H
045E' 0E 00              LD C,0  ; C <= TOTAL NUMBER OF CLUSTERS
0460' 3A 07FD'           LD A,(FSTCLST)
0463' 11 07FF'           LD DE,TBLCLST

0466' 12                RDOPN4: LD (DE),A
0467' 13                 INC DE
0468' FE 7F              CP 7FH
046A' 30 0F              JR NC,RDOPN5
046C' 2A 1F62            LD HL,(_FATBF)
046F' 85                 ADD A,L
0470' 6F                 LD L,A
0471' 30 01              JR NC,RDOPN2
0473' 24                 INC H
0474' 7E                RDOPN2: LD A,(HL)  ;HL = (_FATBF) + (FSTCLST)
0475' 05                 DEC B
0476' 28 24              JR Z,RDOPN3 ; MORE THAN 16 CLUSTER
0478' 0C                 INC C
0479' 18 EB              JR RDOPN4

047B' 0D                RDOPN5: DEC C  ;LAST CLUSTER IS DUMMY
047C' F5                 PUSH AF
047D' 79                 LD A,C
047E' 87                 ADD A,A
047F' 87                 ADD A,A
0480' 87                 ADD A,A
0481' 87                 ADD A,A
0482' 4F                 LD C,A  ;C = C * 16
0483' F1                 POP AF
0484' D6 80              SUB 80H  ;RC is 0 origin. So it isn't 7Fh but 80H.
0486' 81                 ADD A,C
0487' 32 0816'           LD (RC),A  ;RC = Total number of Records

048A' AF                 XOR A
048B' 32 0811'           LD (FLPNT),A
048E' 01 0038            LD BC,InfSize
0491' ED 5B 07DC'        LD DE,(FCB_ADR)
0495' 21 07DF'           LD HL,FILE_BF
0498' ED B0              LDIR
049A' B7                 OR A  ;CY = 0
049B' C9                 RET

049C' 3E 07             RDOPN3: LD A,7  ;Bad Allocation Error
049E' 37                 SCF
049F' C9                 RET

04A0' 3A 1F5D           ROPEN: LD A,(_DSK)
04A3' CD 04BF'           CALL DEVCHK
04A6' D8                 RET C  ;Bad File descripter
04A7' CD 04CA'           CALL FCBSCH
04AA' D8                 RET C
04AB' 3E 08              LD A,8  ;File Not Fonnd
04AD' 37                 SCF
04AE' C0                 RET NZ
04AF' E5                 PUSH HL
04B0' ED 5B 1F74         LD DE,(_IBFAD)
04B4' 01 0020            LD BC,20H
04B7' ED B0              LDIR
04B9' E1                 POP HL
04BA' 7E                 LD A,(HL)
04BB' CD 0514'           CALL FMCHK
04BE' C9                 RET

04BF' FE 41             DEVCHK: CP 'A'
04C1' 38 04              JR C,DEVCH1
04C3' FE 45              CP 'D'+1
04C5' 3F                 CCF
04C6' D0                 RET NC

04C7' 3E 03             DEVCH1: LD A,3 ;Bad File descripter
04C9' C9                 RET
                        ;
                        ; FCB SEARCH
04CA' 0E 10             FCBSCH: LD C,16  ;Directory Lengh
04CC' ED 5B 1F60         LD DE,(_DIRPS) ;Directory start
04D0' ED 53 0812'       FCBSC1: LD (DEBUF),DE
04D4' 2A 1F64            LD HL,(_DTBUF)
04D7' 3E 01              LD A,1
04D9' CD 2000            CALL _DRDSB
04DC' D8                 RET C
04DD' 06 08              LD B,8
04DF' 22 0814'          FCBSC2: LD (HLBUF),HL
04E2' 7E                 LD A,(HL)
04E3' FE FF              CP 0FFH
04E5' 28 1A              JR Z,FCBSC4
04E7' B7                 OR A
04E8' 28 0B              JR Z,FCBSC3
04EA' D5                 PUSH DE
04EB' ED 5B 1F74         LD DE,(_IBFAD)
04EF' CD 0505'           CALL FCOMP
04F2' D1                 POP DE
04F3' 28 0D              JR Z,FCBSC5
04F5' D5                FCBSC3: PUSH DE
04F6' 11 0020            LD DE,32
04F9' 19                 ADD HL,DE
04FA' D1                 POP DE
04FB' 10 E2              DJNZ FCBSC2
04FD' 13                 INC DE
04FE' 0D                 DEC C
04FF' 20 CF              JR NZ,FCBSC1

0501' 3E                FCBSC4: DB 3EH ; Z = 0
0502' AF                FCBSC5: XOR A ; Z = 1
0503' B7                 OR A
0504' C9                 RET

                        ; File Name Compare
0505' C5                FCOMP: PUSH BC
0506' E5                 PUSH HL
0507' 06 10              LD B,16 ;Directory lengh
0509' 13                FCOMP1: INC DE
050A' 23                 INC HL
050B' 1A                 LD A,(DE)
050C' BE                 CP (HL)
050D' 20 02              JR NZ,FCOMP2
050F' 10 F8              DJNZ FCOMP1
0511' E1                FCOMP2: POP HL
0512' C1                 POP BC
0513' C9                 RET

                        ; FILE MODE CHECK
0514' E5                FMCHK: PUSH HL
0515' E6 87              AND 87H  ;10000111B
0517' 21 291F            LD HL,291FH ;%FTYPE
051A' BE                 CP (HL)
051B' E1                 POP HL
051C' C8                 RET Z
051D' 3E 06              LD A,6  ;Bad File Mode
051F' 37                 SCF
0520' C9                 RET
                        ;=====================
                        ; FILE OPEN FOR WRITE
                        ;=====================
0521'                   _WROPEN:
0521' 22 07DC'           LD (FCB_ADR),HL
0524' CD 1FA3            CALL _FILE
0527' D8                 RET C
0528' CD 1FAF            CALL _WOPEN
052B' D8                 RET C
                         ;
052C' 01 0020            LD BC,20H
052F' 11 07DF'           LD DE,FILE_BF
0532' 2A 1F74            LD HL,(_IBFAD)
0535' ED B0              LDIR
0537' 3A 1F5D            LD A,(_DSK)
053A' 32 0810'           LD (FLDSK),A
053D' 2A 27E1            LD HL,(27E1H)
0540' 22 0814'           LD (HLBUF),HL
0543' 2A 27DF            LD HL,(27DFH)
0546' 22 0812'           LD (DEBUF),HL
0549' 21 0000            LD HL,0
054C' 22 07F1'           LD (FLSIZE),HL
                         ;CALL RDFAT  ;FAT has already loaded.
054F' CD 07A6'           CALL FCGET
0552' D8                 RET C

0553' 32 07FD'           LD (FSTCLST),A
0556' 32 07FF'           LD (TBLCLST),A
0559' 3E 80              LD A,80H
055B' 32 0800'           LD (TBLCLST+1),A ;Debugged '90 Feb.16th
055E' AF                 XOR A
055F' 32 0816'           LD (RC),A
0562' 32 07DE'           LD (LST_DSK),A ;Never beibg then same
0565' 32 0811'           LD (FLPNT),A
0568' 21 0000            LD HL,0
056B' 22 07F1'           LD (FLSIZE),HL
056E' 01 0020            LD BC,20H
0571' ED 5B 0814'        LD DE,(HLBUF)
0575' 2A 1F74            LD HL,(_IBFAD)
0578' ED B0              LDIR
057A' 2A 0814'           LD HL,(HLBUF)
057D' 36 05              LD (HL),5  ;Attribute of 'Now Writing...'
057F' 3E 01              LD A,1
0581' ED 5B 0812'        LD DE,(DEBUF)
0585' 2A 1F64            LD HL,(_DTBUF)
0588' CD 2003            CALL _DWTSB
058B' D8                 RET C

058C' 01 0038            LD BC,InfSize
058F' ED 5B 07DC'        LD DE,(FCB_ADR)
0593' 21 07DF'           LD HL,FILE_BF
0596' ED B0              LDIR
0598' B7                 OR A  ;CY = 0
0599' C9                 RET
                        ;==============
                        ; FILE CLOSE
                        ;==============
059A' EB                _CLOSE: EX DE,HL
059B' 11 07DF'           LD DE,FILE_BF
059E' 01 0038            LD BC,InfSize
05A1' ED B0              LDIR
05A3' 3A 0810'           LD A,(FLDSK)
05A6' 32 1F5D            LD (_DSK),A
05A9' 3A 0816'           LD A,(RC)
05AC' 2A 07F1'           LD HL,(FLSIZE)
05AF' 2C                 INC L
05B0' 2D                 DEC L
05B1' 20 01              JR NZ,COL3
05B3' 3C                 INC A
05B4' 67                COL3: LD H,A
05B5' 22 07F1'           LD (FLSIZE),HL
05B8' 3E 01              LD A,1
05BA' ED 5B 0812'        LD DE,(DEBUF)
05BE' 2A 1F64            LD HL,(_DTBUF)
05C1' CD 2000            CALL _DRDSB
05C4' D8                 RET C

05C5' 21 07DF'           LD HL,FILE_BF
05C8' ED 5B 0814'        LD DE,(HLBUF)
05CC' 01 0020            LD BC,20H
05CF' ED B0              LDIR
05D1' 3E 01              LD A,1
05D3' ED 5B 0812'        LD DE,(DEBUF)
05D7' 2A 1F64            LD HL,(_DTBUF)
05DA' CD 2003            CALL _DWTSB
05DD' D8                 RET C
```

▶４年半ぶりに日本に帰ってきました。X68000XVIが発売となり，のどから手が出るほどほしいです。しかし，８月頃に留学のための試験があるのでしばらく我慢します。この年になってまで禁コン生活とは。

野田　直人(33)愛知県

```
05DE' 06 10         LD B,10H
05E0' 21 07FF'      LD HL,TBLCLST
05E3' 7E           COL1: LD A,(HL)
05E4' FE 7F         CP 7FH
05E6' 30 12         JR NC,COL2
05E8' 23            INC HL
05E9' 4E            LD C,(HL)
05EA' E5            PUSH HL
05EB' 2A 1F62       LD HL,(_FATBF)
05EE' 16 00         LD D,0
05F0' 5F            LD E,A
05F1' 19            ADD HL,DE
05F2' 71            LD (HL),C
05F3' E1            POP HL

05F4' 05            DEC B
05F5' CA 049C'      JP Z,RDOPN3 ;Bad File Allocation
05F8' 18 E9         JR COL1

05FA' CD 0795'     COL2: CALL WRFAT
05FD' C9            RET

05FE' 01 0038      _READ:: LD BC,InfSize
0601' EB            EX DE,HL
0602' 11 07DF'      LD DE,FILE_BF
0605' ED B0         LDIR

0607' 3A 0810'      LD A,(FLDSK)
060A' 32 1F5D       LD (_DSK),A
060D' 3A 0811'      LD A,(FLPNT)
0610' 47            LD B,A
0611' 3A 0816'      LD A,(RC)
0614' B8            CP B
0615' DA 049C'      JP C,RDOPN3 ;Bad Allocation File
0618' 78            LD A,B
0619' CD 075B'      CALL PNTREC
061C' EB            EX DE,HL  ; DE = RECORD
061D' D5            PUSH DE
061E' DD E5         PUSH IX
0620' E1            POP HL
0621' 11 003F       LD DE,FCBSIZE
0624' 19            ADD HL,DE
0625' D1            POP DE
0626' 3E 01         LD A,1
0628' CD 2000       CALL _DRDSB
062B' C9            RET

062C'              _tmpREAD::
062C' 01 0038       LD BC,InfSize
062F' 22 07DC'      LD (FCB_ADR),HL
0632' 11 07DF'      LD DE,FILE_BF
0635' ED B0         LDIR
0637' 3A 0810'      LD A,(FLDSK)
063A' 32 1F5D       LD (_DSK),A
063D' 3A 0811'      LD A,(FLPNT)
0640' 47            LD B,A
0641' 3A 0816'      LD A,(RC)
0644' B8            CP B
0645' 38 1A         JR C,sqREAD1 ;Bad Allocation File
0647' 78            LD A,B
0648' CD 075B'      CALL PNTREC
064B' EB            EX DE,HL  ; DE = RECORD
064C' 3E 01         LD A,1
064E' 21 0000*      LD HL,zzbuf
0651' CD 2000       CALL _DRDSB
0654' 38 0B         JR C,sqREAD1
0656' 2A 07DC'      LD HL,(FCB_ADR)
0659' 11 0031       LD DE,31H
065C' 34            INC (HL)
065D' 21 0000       LD HL,0
0660' C9            RET

0661' 21 FFFF      sqREAD1:LD HL,-1
0664' C9            RET

0665' 01 0038      _WRITE::LD BC,InfSize
0668' EB            EX DE,HL  ;HL = FCB ADR
0669' 22 07DC'      LD (FCB_ADR),HL
066C' 11 07DF'      LD DE,FILE_BF
066F' ED B0         LDIR

0671' 3A 0810'      LD A,(FLDSK)
0674' 32 1F5D       LD (_DSK),A
0677' CD 06DE'      CALL SetRC
067A' D8            RET C

067B' 3A 0811'      LD A,(FLPNT)
067E' CD 075B'      CALL PNTREC
0681' EB            EX DE,HL  ;DE = RECORD NO.

0682' D5            PUSH DE
0683' DD E5         PUSH IX
0685' E1            POP HL
0686' 11 003F       LD DE,FCBSIZE
0689' 19            ADD HL,DE
068A' D1            POP DE

068B' 3E 01         LD A,1
068D' CD 2003       CALL _DWTSB
0690' D8            RET C

0691' 01 0038       LD BC,InfSize
0694' ED 5B 07DC'   LD DE,(FCB_ADR)
0698' 21 07DF'      LD HL,FILE_BF
069B' ED B0         LDIR
069D' B7            OR A  ;CY = 0
069E' C9            RET

069F'              _sqWRITE::
069F' 01 0038       LD BC,InfSize
06A2' 22 07DC'      LD (FCB_ADR),HL
06A5' 11 07DF'      LD DE,FILE_BF
06A8' ED B0         LDIR

06AA' 3A 0810'      LD A,(FLDSK)
06AD' 32 1F5D       LD (_DSK),A
06B0' CD 06DE'      CALL SetRC
06B3' 38 25         JR C,sqWRITE1
06B5' 3A 0811'      LD A,(FLPNT)
06B8' CD 075B'      CALL PNTREC
06BB' EB            EX DE,HL  ;DE = RECORD NO.
06BC' 3E 01         LD A,1
06BE' 21 0000*      LD HL,zzbuf
06C1' CD 2003       CALL _DWTSB
06C4' 38 14         JR C,sqWRITE1

06C6' 21 0811'      LD HL,FLPNT
06C9' 34            INC (HL)

06CA' 01 0038       LD BC,InfSize
06CD' ED 5B 07DC'   LD DE,(FCB_ADR)
06D1' 21 07DF'      LD HL,FILE_BF
06D4' ED B0         LDIR
06D6' 21 0000       LD HL,0
06D9' C9            RET

06DA'              sqWRITE1:
06DA' 21 FFFF       LD HL,-1
06DD' C9            RET
                   ;
                   ; Set RC
06DE' 3A 0811'     SetRC: LD A,(FLPNT)
06E1' 47            LD B,A
06E2' 3A 0816'      LD A,(RC)
06E5' B8            CP B
06E6' D0            RET NC

06E7' 3A 0811'      LD A,(FLPNT)
06EA' E6 F0         AND 0F0H
06EC' 47            LD B,A
06ED' 3A 0816'      LD A,(RC)
06F0' E6 F0         AND 0F0H
06F2' B8            CP B
06F3' 30 48         JR NC,SetRC1

06F5' CB 3F         SRL A
06F7' CB 3F         SRL A
06F9' CB 3F         SRL A
06FB' CB 3F         SRL A  ;A = A / 16
06FD' 21 0800'      LD HL,TBLCLST+1
0700' 16 00         LD D,0
0702' 5F            LD E,A
0703' 19            ADD HL,DE

0704' CD 0777'      CALL RDFAT
0707' D8            RET C

0708' E5            PUSH HL

0709' 3A 0816'      LD A,(RC)
070C' E6 F0         AND 0F0H
070E' CD 075B'      CALL PNTREC
0711' CD 07C8'      CALL RECCL

0714' 2A 1F62       LD HL,(_FATBF)
0717' 16 00         LD D,0
0719' 5F            LD E,A  ;A = RC && 0F0H
071A' 19            ADD HL,DE
071B' 36 8F         LD (HL),8FH ;DUMMY
071D' CD 07A6'      CALL FCGET
0720' 77            LD (HL),A
0721' E1            POP HL
0722' D8            RET C
0723' 77            LD (HL),A

0724' 2A 1F62       LD HL,(_FATBF)
0727' 16 00         LD D,0
0729' 5F            LD E,A
072A' 19            ADD HL,DE
072B' 36 80         LD (HL),80H
072D' 3A 0816'      LD A,(RC)
0730' E6 F0         AND 0F0H
0732' C6 10         ADD A,16  ;ONE CLUSTER ADDED
0734' 32 0816'      LD (RC),A

0737' CD 0795'      CALL WRFAT
073A' D8            RET C
073B' 18 A1         JR SetRC

073D' 3A 0811'     SetRC1: LD A,(FLPNT)
0740' 32 0816'      LD (RC),A
0743' CB 3F         SRL A
0745' CB 3F         SRL A
0747' CB 3F         SRL A
0749' CB 3F         SRL A ; A = A / 16
074B' 21 0800'      LD HL,TBLCLST+1
074E' 16 00         LD D,0
0750' 5F            LD E,A
0751' 19            ADD HL,DE
0752' 3A 0811'      LD A,(FLPNT)
0755' E6 0F         AND 0FH
0757' C6 80         ADD A,80H
0759' 77            LD (HL),A
075A' C9            RET   ;RET with CY = 0

                   ; FILE POINTER (A) => RECORD NO. (HL)
075B' F5           PNTREC: PUSH AF
075C' F5            PUSH AF
075D' CB 3F         SRL A
075F' CB 3F         SRL A
0761' CB 3F         SRL A
0763' CB 3F         SRL A  ;A = A / 16
0765' 21 07FF'      LD HL,TBLCLST
0768' 16 00         LD D,0
076A' 5F            LD E,A
076B' 19            ADD HL,DE
076C' 7E            LD A,(HL)
076D' CD 07C0'      CALL CLREC
0770' F1            POP AF
0771' E6 0F         AND 0FH
0773' 85            ADD A,L
0774' 6F            LD L,A
0775' F1            POP AF
0776' C9            RET

                   ; FAT READ TO BUFFER
0777' D5           RDFAT: PUSH DE
0778' E5            PUSH HL
0779' 3A 07DE'      LD A,(LST_DSK)
077C' 57            LD D,A
077D' 3A 1F5D       LD A,(_DSK)
0780' BA            CP D
0781' 28 0F         JR Z,RDFAT_1
0783' 32 07DE'      LD (LST_DSK),A
0786' 3E 01         LD A,1
0788' ED 5B 1F5E    LD DE,(_FATPOS)
078C' 2A 1F62       LD HL,(_FATBF)
078F' CD 2000       CALL _DRDSB
0792' E1           RDFAT_1:POP HL
0793' D1            POP DE
0794' C9            RET

                   ; FAT WRITE FROM BUFFER
0795' D5           WRFAT: PUSH DE
0796' E5            PUSH HL
0797' 3E 01         LD A,1
0799' ED 5B 1F5E    LD DE,(_FATPOS)
079D' 2A 1F62       LD HL,(_FATBF)
07A0' CD 2003       CALL _DWTSB
07A3' E1            POP HL
07A4' D1            POP DE
07A5' C9            RET

                   ; FREE CLUSTER POSITION GET
07A6' C5           FCGET: PUSH BC
07A7' E5            PUSH HL
07A8' 06 80         LD B,80H
07AA' 2A 1F62       LD HL,(_FATBF)
07AD' 7E           FCGET2: LD A,(HL)
07AE' B7            OR A
07AF' 28 08         JR Z,FCGET3
07B1' 23            INC HL
07B2' 10 F9         DJNZ FCGET2
07B4' 3E 09         LD A,9  ;Device Full
07B6' 37            SCF
07B7' 18 04         JR FCGET4
07B9' 3E 80        FCGET3: LD A,80H
07BB' 90            SUB B
07BC' B7            OR A  ;CY = 0
07BD' E1           FCGET4: POP HL
07BE' C1            POP BC
07BF' C9            RET

                   ; CLUSTER (A) => RECORD (HL)
07C0' 26 00        CLREC: LD H,0
07C2' 6F            LD L,A
07C3' 29            ADD HL,HL
07C4' 29            ADD HL,HL
07C5' 29            ADD HL,HL
07C6' 29            ADD HL,HL
07C7' C9            RET

                   ; RECORD (HL) => CLUSTER (A)
07C8' E5           RECCL: PUSH HL
07C9' CB 3C         SRL H
07CB' CB 1D         RR L ;HL/2
07CD' CB 3C         SRL H
07CF' CB 1D         RR L ;HL/4
07D1' CB 3C         SRL H
07D3' CB 1D         RR L ;HL/8
07D5' CB 3C         SRL H
07D7' CB 1D         RR L ;HL/16
07D9' 7D            LD A,L
07DA' E1            POP HL
07DB' C9            RET

07DC'              FCB_ADR:DS 2
07DE'              LST_DSK:DS 1

07DF'              FILE_BF:DS 12H
07F1'              FLSIZE: DS 2    ;(12H) File Size.
07F3'              FLDTADR:DS 2    ;(14H) Start Address.
07F5'              FLEXADR:DS 2    ;(16H) Exec Address.
07F7'               DS 6
07FD'              FSTCLST:DS 1    ;(1EH) First Cluster.
07FE'               DS 1
07FF'              TBLCLST:DS 11H  ;(20H) Cluster table.
0810'              FLDSK: DS 1     ;(31H) The Login disk.
0811'              FLPNT: DS 1     ;(32H) The FILE Pointor.
0812'              DEBUF: DS 2     ;(33H) Record No. Which Have The DIR.
0814'              HLBUF: DS 2 ;(35H) Address Where On INFORMATION.
0816'              RC: DS 1 ;(37H) The Number of Records The File have.

0817'              DFLTDSK:DS 1 ; drive to use if no drive is named
0818'              unit: DS 2 ; I/O structure address to act on
081A'              CHP: DS 2 ;char *chp;
081C'              FILE: DS 2 ;file name
081E'              MODE: DS 2 ;char *mode;(read or write)
0820'              ZCH: DS 2 ;char ch;

0822'              FN: DS 2 ;int fn; i/o function (for cpmio)
0824'              NUBUFF: DS 2 ;# TEMPORARY BUFFER STORAGE
0826'              AFLAG: DS 1 ;# append flag
                   ;
0827'              SVCHP: DS 2 ;char *svchp; saved character pointer
0829'              RSTDOUT:DS 2 ;int rstdout; unit of redirected stdout
                   ;
082B'              zzmem:: DS 2 ;memory pointer used by calloc
082D'              zzstak::DS 2
                   ;
                   ;
                    END ULINK

                        END     ULINK
```

リスト8 POLL.MAC

```
1FD0          getky   EQU     1FD0H
0003          CTRL_C  EQU     3
0013          PAUSE   EQU     19

0000' poll::
0000'   C1            POP     BC
0001'   E1            POP     HL
0002'   E5            PUSH    HL
0003'   C5            PUSH    BC
0004'   7C            LD      A,H
0005'   B5            OR      L
0006'   CA 0032'      JP      Z,CC003
0009'   CD 1FD0       CALL    getky
000C'   FE 13         CP      PAUSE
000E'   20 12         JR      NZ,CC004
0010' CC005:
0010'   CD 1FD0       CALL    getky
0013'   A7            AND     A
0014'   20 FA         JR      NZ,CC005
0016'   21 0000       LD      HL,0
0019'   FE 03         CP      CTRL_C
001B'   C0            RET     NZ
001C'   E5            PUSH    HL
001D'   CD 0000*      CALL    abort
0020'   C1            POP     BC
0021'   C9            RET

0022' CC004:
0022'   FE 03         CP      CTRL_C
0024'   20 08         JR      NZ,CC008
0026'   21 0000       LD      HL,0
0029'   E5            PUSH    HL
002A'   CD 0000*      CALL    abort
002D'   C1            POP     BC
002E' CC008:
002E'   26 00         LD      H,0
0030'   6F            LD      L,A
0031'   C9            RET

0032' CC003:
0032'   CD 1FD0       CALL    getky   ;when pause = 0
0035'   26 00         LD      H,0
0037'   6F            LD      L,A
0038'   C9            RET
      EXT     abort
      END
```

リスト9 DEBUG.MAC

```
              ORG     6000H
6000    01 0021       LD      BC,PeekEnd - PEEK + 1
6003    11 1531       LD      DE,1531H
6006    21 601F       LD      HL,PEEK
6009    ED B0         LDIR
600B    01 0024       LD      BC,PokeEnd - POKE + 1
600E    11 1C00       LD      DE,1C00H
6011    21 603F       LD      HL,POKE
6014    ED B0         LDIR

6016    21 1C00       LD      HL,1C00H
6019    22 1F9B       LD      (1F9BH),HL
601C    C3 1FFA       JP      1FFAH

601F  PEEK:
601F    E5            PUSH    HL

6020    F3            DI
6021    3E 01         LD      A,1
6023    D3 F7         OUT     (0F7H),A
6025    DB E8         IN      A,(0E8H)
6027    CB FF         SET     7,A
6029    CB B7         RES     6,A
602B    D3 E8         OUT     (0E8H),A
602D    CB FC         SET     7,H
602F    CB F4         SET     6,H
6031    7E            LD      A,(HL)

6032    6F            LD      L,A     ;PUSH   A
6033    DB E8         IN      A,(0E8H)
6035    CB BF         RES     7,A
6037    CB F7         SET     6,A
6039    D3 E8         OUT     (0E8H),A
603B    7D            LD      A,L     ;POP    A
603C    FB            EI

603D    E1            POP     HL
603E    C9            RET

603F  PeekEnd:

603F  POKE:
603F    F5            PUSH    AF
6040    C5            PUSH    BC
6041    E5            PUSH    HL

6042    F3            DI
6043    47            LD      B,A     ;Save   Data
6044    3E 01         LD      A,1
6046    D3 F7         OUT     (0F7H),A
6048    DB E8         IN      A,(0E8H)
604A    CB FF         SET     7,A
604C    CB B7         RES     6,A
604E    D3 E8         OUT     (0E8H),A

6050    CB FC         SET     7,H
6052    CB F4         SET     6,H
6054    70    LD      (HL),B

6055    DB E8         IN      A,(0E8H)
6057    CB BF         RES     7,A
6059    CB F7         SET     6,A
605B    D3 E8         OUT     (0E8H),A
605D    FB    EI

605E    E1    POP     HL
605F    C1    POP     BC
6060    F1    POP     AF
6061    C9    RET

6062  PokeEnd:

      END
```

▶「パロディウスだ！」はすごいね，いいね，やったね。「アフターバーナー」以来のスマッシュヒットだと思う。ほかのメーカーもこれから大変だねー。コナミは偉い！

阿部　哲也(17)兵庫県

全機種共通 システムインデックス

■85年 6 月号
序論　共通化の試み
第 1 部　S-OS"MACE"
第 2 部　Lisp-85インタプリタ
第 3 部　チェックサムプログラム
■85年 7 月号
第 4 部　マシン語プログラム開発入門
第 5 部　エディタアセンブラZEDA
第 6 部　デバッガツールZAID
■85年 8 月号
第 7 部　ゲーム開発パッケージBEMS
第 8 部　ソースジェネレータZING
■85年 9 月号
インタラプト　S-OS番外地
第 9 部　マシン語入力ツールMACINTO-S
第10部　Lisp-85入門(1)
■85年10月号
第11部　仮想マシンCAP-X85
連載　Lisp-85入門(2)
■85年11月号
連載　Lisp-85入門(3)
■85年12月号
第12部　Prolog-85発表
■86年 1 月号
第13部　リロケータブルのお話
第14部　FM音源サウンドエディタ
■86年 2 月号
第15部　S-OS"SWORD"
第16部　Prolog-85入門(1)
■86年 3 月号
第17部　magiFORTH発表
連載　Prolog-85入門(2)
■86年 4 月号
第18部　思考ゲームJEWEL
第19部　LIFE GAME
連載　基礎からのmagiFORTH
連載　Prolog-85入門(3)
■86年 5 月号
第20部　スクリーンエディタE-MATE
連載　実戦演習magiFORTH
■86年 6 月号
第21部　Z80TRACER
第22部　magiFORTH TRACER
第23部　ディスクダンプ&エディタ
第24部　"SWORD" 2000 QD
連載　対話で学ぶ magiFORTH
特別付録　PC-8801版S-OS"SWORD"
■86年 7 月号
第25部　FM音源ミュージックシステム
付録　FM音源ボードの製作
連載　計算力アップのmagiFORTH
特別付録　SMC-777版 S-OS"SWORD"
■86年 8 月号
第26部　対局五目並べ
第27部　MZ-2500版 S-OS"SWORD"
■86年 9 月号
第28部　FuzzyBASIC 発表
連載　明日に向かって magiFORTH
■86年10月号
第29部　ちょっと便利な拡張プログラム
第30部　ディスクモニタ DREAM
第31部　FuzzyBASIC 料理法〈1〉
■86年11月号
第32部　パズルゲーム HOTTAN
第33部　MAZE in MAZE
連載　FuzzyBASIC 料理法〈2〉
■86年12月号
第34部　CASL & COMET
連載　FuzzyBASIC 料理法〈3〉
■87年 1 月号
第35部　マシン語入力ツールMACINTO-C
連載　FuzzyBASIC 料理法〈4〉
■87年 2 月号
第36部　アドベンチャーゲーム MARMALADE
第37部　テキアペ作成ツール CONTEX
■87年 3 月号
第38部　魔法使いはアニメがお好き
第39部　アニメーションツール MAGE
付録　"SWORD" 再掲載と MAGIC の標準化
■87年 4 月号
第40部　INVADER GAME
第41部　TANGERINE
■87年 5 月号
第42部　S-OS"SWORD" 変身セット
第43部　MZ-700用 "SWORD" を QD 対応に
■87年 6 月号
インタラプト　コンパイラ物語
第44部　FuzzyBASIC コンパイラ
第45部　エディタアセンブラ ZEDA-3
■87年 7 月号
第46部　STORY MASTER
■87年 8 月号
第47部　パズルゲーム碁石拾い
第48部　漢字出力パッケージ JACKWRITE
特別付録　FM-7/77版 S-OS"SWORD"
■87年 9 月号
第49部　リロケータブル逆アセンブラ Inside-R
特別付録　PC-8001/8801 版 S-OS"SWORD"
■87年10月号
第50部　tiny CORE WARS
第51部　FuzzyBASIC コンパイラの拡張
第52部　X1turbo 版 S-OS"SWORD"
■87年11月号
序論　神話のなかのマイクロコンピュータ
付録　S-OS の仲間たち
第53部　もうひとつの FuzzyBASIC 入門
第54部　ファイルアロケータ&ローダ
インタラプト　S-OS こちら集中治療室
第55部　BACK GAMMON
■87年12月号
第56部　タートルグラフィックパッケージTURTLE
第57部　X1turbo 版 "SWORD" アフターケア
ラインプリントルーチン
特別付録　PASOPIA7 版 S-OS"SWORD"
■88年 1 月号
第58部　FuzzyBASIC コンパイラ・奥村版
付録　石上版コンパイラ拡張部の修正
■88年 2 月号
第59部　シューティングゲーム ELFES
■88年 3 月号
第60部　構造型コンパイラ言語 SLANG
■88年 4 月号
第61部　デバッギングツール TRADE
第62部　シミュレーションウォーゲーム WALRUS
■88年 5 月号
第63部　シューティングゲーム ELFES II
第64部　地底最大の作戦
■88年 6 月号
第65部　構造化言語 SLANG 入門(1)
第66部　Lisp-85 用 NAMPA シミュレーション
■88年 7 月号
第67部　マルチウィンドウドライバ MW-1
連載　構造化言語 SLANG 入門(2)
■88年 8 月号
第68部　マルチウィンドウエディタ WINER
■88年 9 月号
第69部　超小型エディタ TED-750
第70部　アフターケア WINER の拡張
■88年10月号
第71部　SLANG 用ファイル入出力ライブラリ
第72部　シューティングゲーム MANKAI
■88年11月号
第73部　シューティングゲーム ELFES Ⅳ
■88年12月号
第74部　ソースジェネレータ SOURCERY
■89年 1 月号
第75部　パズルゲーム LAST ONE
第76部　ブロックゲーム FLICK
■89年 2 月号
第77部　高速エディタアセンブラ REDA
特別付録　X1版 S-OS"SWORD"〈再掲載〉
■89年 3 月号
第78部　Z80用浮動小数点演算パッケージSOROBAN
■89年 4 月号
第79部　SLANG 用実数演算ライブラリ
■89年 5 月号
第80部　ソースジェネレータ RING
■89年 6 月号
第81部　超小型コンパイラTTC
■89年 7 月号
第82部　TTC用パズルゲーム　TICBAN
■89年 8 月号
第83部　CP/M用ファイルコンバータ
■89年 9 月号
第84部　生物進化シミュレーションBUGS
■89年10月号
第85部　小型インタプリタ言語TTI
■89年11月号
第86部　TTI用パズルゲーム　PUSH BON!
■89年12月号
第87部　SLANG用リダイレクションライブラリ
DIO. LIB
■90年 1 月号
第88部　SLANG用ゲームWORM KUN
特別付録　再掲載SLANGコンパイラ
■90年 2 月号
第89部　超小型コンパイラTTC++
■90年 3 月号
第90部　超多機能アセンブラOHM-Z80
■90年 4 月号
第91部　ファジィコンピュータシミュレーションI-MY
■90年 5 月号
第92部　インタプリタ言語STACK
■90年 6 月号
第93部　リロケータブルフォーマットの取り決め
第94部　STACK用ゲーム　SQUASH！
第95部　X68000対応S-OS "SWORD"
特別付録　PC-286対応S-OS "SWORD"
■90年 7 月号
第96部　リロケータブルアセンブラWZD
■90年 8 月号
第97部　リンカWLK
■90年 9 月号
第98部　BILLIARDS
■90年10月号
第99部　ライブラリアンWLB
■90年11月号
第100部　タブコード対応エディタEDC-T
■90年12月号
第101部　STACKコンパイラ
■91年 1 月号
第102部　ブロックアクションゲーム COLUMNS
■91年 2 月号
第103部　ダイスゲームKISMET
■91年 3 月号
第104部　アクションゲームMUD BALLIN'
■91年 4 月号
第105部　SLANG用カードゲームDOBON
■91年 5 月号
第106部　実数型コンパイラ言語REAL

＊以上のアプリケーションは，基本システムである S-OS "MACE" または S-OS "SWORD" がないと動作しませんのでご注意ください。

INDEX

▶今日，朝起きるのが早すぎたので沼を作って時間をつぶしてから学校へ行こうと思いました。さて，1ゲーム終わって時計を見ると……ぎえ～！　高校ではこんなこと考えられなかったのに。ああ，大学ってお茶目。　大平　浩貴(18)埼玉県

Oh!X L・I・V・E in '91

X68000用 暴れん坊将軍より夜明け
Nishikawa Zenji
西川 善司

X68000用 ふしぎの海のナディアより ブルーウォーター
Kanou Shinya/Ohara Yoshinori
加納 伸也/小原 良宣

X68000用 POWER HOLL
Amano Takayuki
天野 貴之

X1/turbo用 魔法の妖精ペルシャより 見知らぬ国のトリッパー
Katou Takashi
加藤 隆

今月は9周年特別企画（？）というわけで，イロモノ一直線でお届けします。まあ，たまにはこんなのもいいでしょう。とはいえ，曲のレベルが低くなっているわけではありませんのでご安心を。こういったちょっとくだけたものからミュージックプログラムに慣れていただけると，とってもうれしいのですが……。

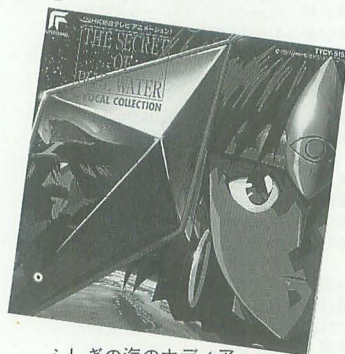
暴れん坊将軍のイースアレンジ？

このコーナーに載るのは久しぶりの，ボンバーマン・キング西川善司であります。

さて，テレビ朝日系火曜～金曜，朝10時半からやっている(もう終わったかも)「暴れん坊将軍III(再)」をご存じですか。休みの日にテレビをつけたら，たまたまやっていたのですが，この番組のオープニングテーマ，結構かっこいいんですな。しかしこの曲調，なにかに似ている……。そうだ！「イースI」の草原のテーマだ。もし，この曲を昔のファルコム風にアレンジしたら面白いかも……。そんな思惑を交錯しながらできたのがリスト1です。ちなみに，**AD PCM音は使用しておりません**のでOPMDやOPMAなどの特殊ドライバは必要ありません。

プログラムは，OPN機種(MZ-2500/PC-8801・9801/FM77)への移植も考えて，基本的にはFM3，PSG3の構成になっています(X1系のOPM機種への移植の場合は，なんの苦労もないでしょう)。

……OPN機種への移植は……

OPN機種へ移植する際にはTR1のメロディ，TR3のベース，TR8のドラムスをFM音源に割り当てるといいでしょう。使用した音色もDT2，LFOなどのOPM専用特殊機能は使っていないので，そのまま持っていけるはずです。あとは，TR4～6のコード3声をPSGに割り当てればメデタシ，メデタシ。余力のある人は，PSGのノイズ機能を使ってTR7のハイハットパートを割り込ませましょう。そのほか，移植に関して気をつけたいのは，X68000のMMLはオクターブスイッチ「<」「>」が従来のものとは正反対という点。

ふしぎの海のナディア

さて，打ち込みや移植が完了したらゲーム好きの友人でも家に呼び寄せて「これ，なんのゲームの曲だ？」と聞いてやりましょう。

ところで，私ってOh!Xでゲームミュージック以外を発表するのって，はじめてじゃないか……？　(善)

椰子からナディア

この春で終了してしまったアニメ「ふしぎの海のナディア」より，「ブルーウォーター」をフルコーラスでお届けしましょう。X68000&OPMD用です。

NHK総合のアニメーションのわりには(失礼)女の子がかわいいので話題になったアレですね。ネットなどのCGデータでも，ときどきお目にかかることができます。

さて，曲の話に移りましょう。うむむ，よくできている。いや，できすぎている。今月は9周年ということもあって，比較的イロモノの予定だったのですが，これじゃオチがとれない(？)ですね。

よくできている作品に限って細かい点が気になるものですが，この作品もいくつか気になった点がありましたので挙げておきましょう。曲の流れとして，押さえている部分が少ないと思います。1つひとつの音やフレーズが素晴らしいのはわかりますが，押さえることもできるようになると常連入りのレベルになるでしょう。原曲と比べるとわかりますよね。まあ，ドラムのボリューム調整がないことを考えると，しかたないことなのでしょうか。

それから，ハードウェアLFO(以下HLFO)を使っていますね。パソコンのMMLでは音色を設定するたびにHLFOの値を設定し直していますので，すべての音色に同じHLFOの値をセットしておかないと，とんでもないことになりますよね。音色の作り方から見て，小原君はそこいらへんを理解しているようですが，ボーカルの音色だけスピードが208になっているのはなぜでしょう。ちょっと気になりました。もし，ちゃんとした理由があるのでしたら後学のために教えてください。

注意点として，サンプリングデータはOPMD用が指定されています。OPMA用でも聴くことはできますが，作者たちのイメージとは違うようです。OPMD用を持っている人はそちらで聴いてください。さらにナディア用のコンフィグファイルを掲載しておきます。OPMDを組み込むときに注意してください。普通に鳴らすとオーケストラヒットの嵐になります。

この作品はとってもいいです。原曲を知らない人でもぜひとも入力しましょう。

※加納君へ。高級ヘッドホン＋延長ケーブルを手に入れて「京間6畳対角システム」に対抗してみてください。ヘッドホンのメーカーはオーディオテクニカをお勧めします。次の作品もお待ちしてます。

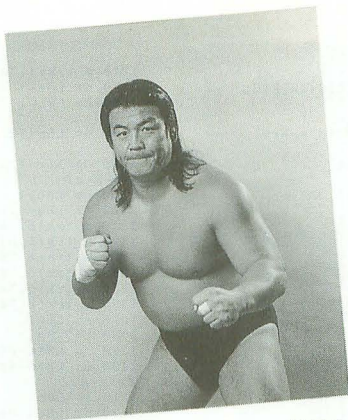
必殺！　サソリ固め

おーっと，いきなり長州力のテーマだあ！　プロレスラーのテーマソングに目をつけるところがにくい。にくい，にくい，にくいあなたはお肉屋さん？　そんなことはどうでもいい，まさにMMLルネッサンス，イロモノ路線の革命戦士が，いまここに現れた〜っ！

古舘伊知郎風に読んでくれた人，ありがとうございます。最近，プロレスというものを見なくなってしまったので，私が見ていた当時の名（迷？）アナウンサーを真似させていただきました。

X68000のOPMD用にお届けしましょう。曲は「POWER HOLL」です。いかにもシンセサイザミュージックというのが私はとっても好きです。この曲はかの名機，初代DX7が登場する前ぐらいの曲ですので，ちょうどYMOや「テクノ」なんて言葉が流行っていた頃のものでしょう。革命戦士・長州力らしく，当時の最先端のシンセサイザを使ったテーマソングですが，いまとなってはあまり使われない音になってしまった感じがしますね。

天野君は初投稿だそうですが，MMLを覚えてはじめて作った曲もこの「POWER HOLL」だそうです。よほど思い入れがあるのでしょう。そういった曲は何度も作り直してみることをお勧めします。昔の自分の作品を聴くと恥ずかしいってぐらいになれば立派なものです。それだけレベルが上がったということですから。

実は，10年ほど前にこの曲のテープを手に入れ，当時放送委員だった私はお昼の校内放送で流したものでした。あの当時のテープを引っ張り出して聴き直してしまいました。曲のデキですが，若干のアレンジが入っているという感じで，長さも短くなっていますね。原曲はフェードアウトだったし。作者の天野君の発見ですが，590行と600行を削るとまた違ったアレンジになるとのこと。途中から崩れてしまうのが難点ですが，確かに面白いですね。

長州力

今回得た教訓，「ドラムを無意味に左右に振ると曲に勢いがなくなる」だそうです。皆さんも参考にしてください。

ペルッコラブリンクルクルリンクル

えっと，X1のMusic BASIC用には「魔法の妖精ペルシャ」より，「見知らぬ国のトリッパー」ですの。でもでもオープニングのバージョンではありませんの。この作品は番組中に流れていたピアノの曲ですの。

なんだかもう好きにしてっ，て感じの書き出しですが，最後までおつきあいください。作者の加藤君は'91年2月号の「リンゴの森の子猫たち」の作者なので，覚えている人もいると思います。ちょっとアニメのナツメロ路線といった感じもありますが，常連目指して頑張ってください。

プログラムを見てみると，もっと短くなりそうでした。曲データを短くすることは打ち込みやすいことにもなりますので，できるだけ考えたほうがよいでしょう。投稿されてきたリストはコメントを入れて550行と，演奏されるデータに対して大きすぎると思われたため，リストの大部分を変更させていただきました。もちろん，スピーカーから出てくる音は同じです。作品に対しての修正はありません。加藤君は1小節ごとにMML化していったようですが，ピアノ曲のようなものは2小節や4小節ごとにMML化していったほうが入力しやすくなると思いますよ。

関係ないのですが，私はティラミスチョコレートを美味しいとは思いません。それから意味不明の参考資料をありがとうございました。それでは，クルクルピカリンクルピカリン。

担当者より

今回の投稿は，皆さんコメントがいっぱい書いてありましたので，とても楽しく読ませていただきました。どうもありがとうございます。これから投稿される人もよろしくお願いいたします。苦労したことや，ちょっとしたテクニック，関係のない話や最近飲んだまずいジュースの話でもかまいません。投稿のついでのコメントでもコメントのついでの投稿でもお待ちしてます。

ところで，LIVE inではゲームミュージック特集を近いうちにやりたいと思っています。ストックの中から常連さんのスーパーな作品をバシバシ掲載する予定ですので楽しみにしていてください。　(S.K.)

リスト1　夜明け

```
 10 m_init():dim char v(4,10)
 20 /*  AF   OM   WF   SY   SP  PMD  AMD  PMS  AMS  PAN
 30 v={ 60,  15,   0,   0,   0,   0,   0,   0,   0,   3,   0,
 40 /*  AR   DR   SR   RR   SL   OL   KS   ML  DT1  DT2  AME
 50      31,   6,   0,   2,   1,  28,   0,   2,   3,   0,   0,
 60      31,   0,   0,   5,   0,   0,   0,   2,   3,   0,   0,
 70      31,   6,   0,   2,   1,  12,   0,   1,   7,   0,   0,
 80      31,   0,   0,   5,   0,   0,   0,   2,   7,   0,   0}
 90 m_vset(2,v)          /*SYNTHE LEAD
100 /*  AF   OM   WF   SY   SP  PMD  AMD  PMS  AMS  PAN
110 v={ 32,  15,   0,   0,   0,   0,   0,   0,   0,   3,   0,
120 /*  AR   DR   SR   RR   SL   OL   KS   ML  DT1  DT2  AME
130      31,   6,   6,   5,   2,  28,   3,  12,   6,   0,   0,
140      31,   5,   5,   5,   1,  58,   3,  10,   6,   0,   0,
150      31,   8,   5,   5,   1,  22,   2,   1,   6,   0,   0,
160      31,   5,   7,   6,  15,   0,   2,   2,   6,   0,   0}
170 m_vset(3,v)          /*E.BASS
180 /*  AF   OM   WF   SY   SP  PMD  AMD  PMS  AMS  PAN
190 v={ 61,  15,   2,   0, 205,  28,   0,   0,   0,   3,   0,
200 /*  AR   DR   SR   RR   SL   OL   KS   ML  DT1  DT2  AME
210      26,  10,   5,  10,  10,   0,   2,  14,   0,   3,   0,
220      31,  20,  11,  15,  11, 127,   1,   3,   7,   1,   0,
230      28,  24,  10,  15,  10,   2,   0,   4,   7,   1,   0,
240      19,  21,  12,  15,  11,   0,   0,   2,   7,   1,   0}
250 m_vset(4,v)          /*CLOSED HH
260 /*  AF   OM   WF   SY   SP  PMD  AMD  PMS  AMS  PAN
270 v={ 61,  15,   0,   0,   0,   0,   0,   0,   0,   3,   0,
280 /*  AR   DR   SR   RR   SL   OL   KS   ML  DT1  DT2  AME
290      31,   0,   0,  15,   0,  28,   0,   2,   3,   0,   0,
300      31,   7,   4,   7,   2,  15,   0,   1,   3,   0,   0,
310      31,   7,   4,   7,   2,  15,   0,   1,   3,   0,   0,
320      31,   7,   4,   7,   2,  15,   0,   1,   3,   0,   0}
330 m_vset(7,v)          /*PSG (LIKE BRASS)
340 /*  AF   OM   WF   SY   SP  PMD  AMD  PMS  AMS  PAN
350 v={ 61,  15,   0,   0,   0,   0,   0,   0,   0,   3,   0,
360 /*  AR   DR   SR   RR   SL   OL   KS   ML  DT1  DT2  AME
370      31,   0,   0,  15,   0,  28,   0,   2,   7,   0,   0,
380      31,  11,   7,   5,   3,  15,   1,   1,   7,   0,   0,
390      31,  11,   7,   5,   3,  15,   1,   1,   7,   0,   0,
400      31,  11,   7,   5,   3,  15,   1,   1,   7,   0,   0}
410 m_vset(8,v)          /*PSG (LIKE BELL)
420 /*  AF   OM   WF   SY   SP  PMD  AMD  PMS  AMS  PAN
430 v={ 59,  15,   0,   0,   0,   0,   0,   0,   0,   3,   0,
440 /*  AR   DR   SR   RR   SL   OL   KS   ML  DT1  DT2  AME
450      22,   0,   0,  10,   0,  13,   0,  10,   0,   0,   0,
460      26,  26,   0,  10,  15,  19,   0,  13,   0,   0,   0,
470      26,  22,   0,  11,  15,  11,   0,   1,   0,   0,   0,
480      30,  14,   0,   7,  15,   0,   1,   1,   0,   0,   1}
490 m_vset(9,v)          /*BASS DRUM
500 /*  AF   OM   WF   SY   SP  PMD  AMD  PMS  AMS  PAN
510 v={ 60,  15,   0,   0,   0,   0,   0,   0,   0,   3,   0,
520 /*  AR   DR   SR   RR   SL   OL   KS   ML  DT1  DT2  AME
530      31,   9,   8,   9,  10,   0,   0,   0,   4,   0,   0,
540      31,  15,  14,   5,   9,   0,   1,   0,   4,   0,   0,
550      31,  20,  18,   8,  11,   0,   1,   0,   4,   0,   0,
560      31,  10,  10,   9,   9,   0,   1,   0,   4,   0,   0}
570 m_vset(10,v)         /*SNARE DRUM
580 /*  AF   OM   WF   SY   SP  PMD  AMD  PMS  AMS  PAN
590 v={ 60,  15,   0,   0,   0,   0,   0,   0,   0,   3,   0,
600 /*  AR   DR   SR   RR   SL   OL   KS   ML  DT1  DT2  AME
610      27,  25,   5,   2,   0,  12,   0,   0,   0,   0,   0,
620      31,  18,  18,   3,   0,   0,   0,   0,   7,   0,   0,
630      31,  20,   0,   0,  12,  12,   3,   1,   0,   0,   0,
640      31,  10,  15,   6,   0,   0,   0,   1,   3,   0,   0}
```

```
 650 m_vset(12,v)       /*TOM TOM
 660 str a[256],b[256],c[256],d[256],e[256],f[256],g[256],h[256
 670 m_tempo(130)
 680 for i=1 to 8:m_alloc(i,6000):next
 690 key 2,"m_play()@M":key 12,"M_STOP()@M":key 19,"SAVE"+chr$(
34)+"abare":key 7,"m_trk("
 700 m_trk(2,"r16."):for i=1 to 8:m_trk(i,"[do]"):next
 710 /*
 720 /*          TRACK 1 (MELODY)          PART 1  (FM 1)
 730 /*
 740 a="L8g4.g<c2 gf4e-d2 c4e-4b-4.<c> g1>
 750 b="L8g4.g<c2 e-d4>b-<c2 g4fe-d4.g c1
 760 c="L8a-4f4.<c>b-a- g4.e-&e-2 a-4f4.fe-f g1
 770 d="L8a-4f4.fe-f g4.c&c2              d4d4db-4a- g1
 780 e=">"+a
 790 f="L8g4.g<c2  e-d4>b-<c2 g4fe-L6d>b-g< c1
 800 m_trk(1,"@2 o4 q8 v14 y48,0"+a)
 810 m_trk(1,b)
 820 m_trk(1,c)
 830 m_trk(1,d)
 840 m_trk(1,e)
 850 m_trk(1,f)
 860 /*
 870 /*          TRACK 2 (MELODY CHORUS) PART 1' (XXX)
 880 /*
 890 m_trk(2,"@2 o4 q8 v10 y49,40"+a)
 900 m_trk(2,b)
 910 m_trk(2,c)
 920 m_trk(2,d)
 930 m_trk(2,e)
 940 m_trk(2,f)
 950 /*
 960 /*          TRACK 3 (BASS)            PART 2  (FM 2)
 970 /*
 980 a="L16|:4crcc:| |:4frff:| |:4grgg:| |:4crcc:|
 990 b="L16|:4crcc:| |:frff:||:e-re-e-:| |:4grgg:| |:4crcc:|
1000 c="L16|:4frff:| |:4crcc:| |:4frff:| |:4grgg:|
1010 d="L16|:4frff:| |:4crcc:| |:4drdd:| |:4grgg:|
1020 e=a
1030 f="L16|:4crcc:| >|:b-rb-b-:||:a-ra-a-:| |:4grgg:| <|:4crcc
:|
1040 m_trk(3,"@3 o2 q8 v13 y50,0"+a)
1050 m_trk(3,b)
1060 m_trk(3,c)
1070 m_trk(3,d)
1080 m_trk(3,e)
1090 m_trk(3,f)
1100 /*
1110 /*          TRACK 4-6 (BACKING)     PART 3,4,5 (PSG 1,2,3)
1120 /*
1130 a="L8g16e-16c16g16&g2. b-2&b-fe-d  e-2d2  c2&cfg4
1140 b="L8g16e-16c16g16&g2. f2e-2  d1   c4L16cde-fga-b-<cde-fg
1150 c="L8f4a-2.    e-g4e-&e-2            f4a-2. q6g4g4g4q8<L16d
e-fg
1160 d="L8f4a-2.    e-g4e-&e-2          >b-1       <g4<L16b-a-rg
a-grfgfe-d
1170 e="L8g16e-16c16g16&g2. b-2&b-fe-d           e-2d2   c2&cfg4
1180 f="L8g4.g<c2  e-d4>b-<c2     g4fe-L6d>b-g< c1
1190 m_trk(4,"@7 o4 q8 v13 y51,20"+a)
1200 m_trk(4,b)
1210 m_trk(4,"@7 o4 q8 v13 y51,20"+c)
1220 m_trk(4,"@7 o4 q8 v13 y51,20"+d)
1230 m_trk(4,"@7 o4 q8 v13 y51,20"+e)
1240 m_trk(4,"@7 o4 q8 v13 y51,20"+f)
1250 a="L8e-1                 f2&fdc>b-  <c2>b-2 g2&g<de-4
1260 b="L8e-1                 d2c2  >b-1  g4<@8L16y52,40cde-fga-b
fg                                                         -<cde-
1270 c="L8c4f2.     ce-4c&c2              c4f2.  q6e-4e-4e-4q8L1
cd                                                          6a-b-<
1280 d="L8c4f2.     ce-4c&c2           >f1         <d4L16<dc>rb-
a-b-a-gf
1290 e="L8e-1                 f2&fdc>b-            g2 b-2 g2&g<de-4
1300 f="L8e-4.e-g2 d2e-2            e-4dc>b-2     g1
1310 m_trk(5,"@7 o4 q8 v13 y52,20"+a)
1320 m_trk(5,b)
1330 m_trk(5,"@7 o4 q8 v13 y52,20"+c)
1340 m_trk(5,"@7 o4 q8 v13 y52,20"+d)
1350 m_trk(5,"@7 o4 q8 v13 y52,20"+e)
1360 m_trk(5,"@7 o4 q8 v13 y52,20"+f)
1370 a="L8c1                  d2&d>b-gf   g2 f2  e-2&e-b-<c4
1380 b="L8c1                  >b-2g2  g1   e-4L16ga-b-<cde-fga-b-<cd
1390 c="L8>a-4<c2.> g<c4>g4<<<y53,40e-dc f2a-2  g2L16>>@8ga-b-<cde-fg
1400 d="L8>a-4<c2.> g<c4>g4<<y53,40gfe- d2&d>b-4a- g4<L16@8b-a-rga-gr
fgfe-d
1410 e="L8c1                  d2&d>b-<<L16gfe-d  c2>>f2  e-4<<e-g8b-g2
1420 f="L8c4.ce-2  >b-2<<cd&e-&f& L24g&g+&g&g+&g&g+&g&g+&g&g+&g&g+>d2
e-1
1430 m_trk(6,"@7 o4 q8 v13 y53,20"+a)
1440 m_trk(6,b)
1450 m_trk(6,"@7 o4 q8 v13 y53,20"+c)
1460 m_trk(6,"@7 o4 q8 v13 y53,20"+d)
1470 m_trk(6,"@7 o4 q8 v13 y53,20"+e)
1480 m_trk(6,"@7 o4 q8 v13 y53,20"+f)
1490 /*
1500 /*          TRACK 7 (HI HAT)          PART 6 (PSG NOISE)
1510 /*
1520 a="L16|:64c:|
1530 b=a
1540 c=a
1550 d=a
1560 e=a
1570 f=a
1580 m_trk(7,"@4 o5 q8 v12 y54,0"+a)
1590 m_trk(7,b)
1600 m_trk(7,c)
1610 m_trk(7,d)
1620 m_trk(7,e)
1630 m_trk(7,f)
1640 /*
1650 /*          TRACK 8 (DRUMS)           PART 7  (FM 3)
1660 /*
1670 a="L4|:7@9o2q8c@10o2c:|@9c@10c8c8
1680 b="L4|:7@9o2q8c@10o2c:|L16ccL8@12o4c>af
1690 c="L4|:6@9o2q8c@10o2c:|ccc@12o4L16c>afc
1700 d="L4|:6@9o2q8c@10o2c:|@9cL16|:@10ccr@9c:|@10cccc
1710 e="L4|:8@9o2q8c@10o2c:|
1720 f="L4|:5@9o2q8c@10o2c:|L6ccc L4|:@9c@10c:|
1730 m_trk(8,"q8 @v127 y55,0"+a)
1740 m_trk(8,b)
1750 m_trk(8,"q8 @v127 y55,0"+c)
1760 m_trk(8,d)
1770 m_trk(8,e)
1780 m_trk(8,f)
1790 for i=1 to 8:m_trk(i,"[loop]"):next:m_play():end
```

リスト2　ブルーウォーター

日本音楽著作権協会(出)許諾第9170217-101号

```
  10 /*
  20 /*「ふしぎの海のナディア」オープニングテーマソング
  30 /*           「　ブルーウォーター　」
  40 /*             唄       森　川　美　穂
  50 /*           作　詞     来　生　えつこ
  60 /*           作　曲     井上　ヨシマサ
  70 /*           編　曲    ジョー＝リノイエ
  80 /*
  90 /* key  4,"1.1420- @M
 100 key 13,"m_play() @M" : key 19,"m_tempo(200) @M
 110 key 14,"m_stop() @M" : key 20,"m_tempo(141) @M
 120 /*
 130 /* E x t e n d e d   V o i c e s
 140 /*
 150 dim char v(4,10)
 160 /*
 170 /* AF  OM  WF  SY  SP PMD AMD PMS AMS PAN
 180 v={60, 15,  2,  0,200, 40,  0,  0,  0,  2,  0,
 190 /* AR  DR  SR  RR  SL  OL  KS  ML DT1 DT2 AME
 200     31,  6,  6,  0,  6,  0,  0, 15,  7,  1,  0,
 210     31, 16,  4, 15, 15, 10,  0,  2,  3,  2,  0,
 220     31, 18,  6,  3,  4,  0,  0,  1,  7,  2,  0,
 230     31, 20,  6, 15, 12,  5,  2,  7,  3,  3,  0}
 240 m_vset(71,v)       /* Hi-Hat Closed ( o5c )
 250 /*
 260 /* AF  OM  WF  SY  SP PMD AMD PMS AMS PAN
 270 v={60, 15,  2,  0,200, 40,  0,  0,  0,  2,  0,
 280 /* AR  DR  SR  RR  SL  OL  KS  ML DT1 DT2 AME
 290     31,  6,  6,  0,  6,  0,  0, 15,  7,  1,  0,
 300     31, 11,  4,  6,  8, 13,  0,  2,  3,  2,  0,
 310     31, 18,  6,  4,  4,  0,  0,  1,  7,  2,  0,
 320     31, 25,  6,  6, 11,  8,  2,  7,  3,  3,  0}
 330 m_vset(72,v)       /* Hi-Hat Open ( o5c )
 340 /*
 350 /* AF  OM  WF  SY  SP PMD AMD PMS AMS PAN
 360 v={51, 15,  2,  0,200, 40,  0,  0,  0,  3,  0,
 370 /* AR  DR  SR  RR  SL  OL  KS  ML DT1 DT2 AME
 380     31,  6,  0,  3, 15, 18,  0,  1,  7,  1,  0,
 390     31,  2,  0,  1, 15, 13,  0,  6,  0,  0,  0,
 400     31, 20,  0, 10, 15, 22,  0,  3,  0,  0,  0,
 410     18, 11,  0,  5, 15,  0,  0, 13,  3,  0,  0}
 420 m_vset(73,v)       /* Clash Cymbal ( o6g )
 430 /*
 440 /* AF  OM  WF  SY  SP PMD AMD PMS AMS PAN
 450 v={50, 15,  2,  0,200, 40,  0,  0,  0,  3,  0,
 460 /* AR  DR  SR  RR  SL  OL  KS  ML DT1 DT2 AME
 470     28,  8,  3,  1,  1, 28,  0,  1,  7,  0,  0,
 480     22, 12,  3,  1,  6, 36,  0,  4,  3,  0,  0,
 490     22, 10,  3,  1,  5, 40,  0,  0,  3,  0,  0,
 500     28,  4,  5,  7,  3,  0,  0,  1,  0,  0,  0}
 510 m_vset(74,v)       /* Synthe Bass
 520 /*
 530 /* AF  OM  WF  SY  SP PMD AMD PMS AMS PAN
 540 v={58, 15,  2,  0,208, 40,  0,  3,  0,  3,  0,
 550 /* AR  DR  SR  RR  SL  OL  KS  ML DT1 DT2 AME
 560     19,  9,  0,  5,  1, 25,  0,  1,  7,  0,  0,
 570     18,  9,  0,  4,  5, 55,  0,  8,  0,  2,  0,
 580     20,  0,  0,  4,  0, 45,  0,  1,  0,  0,  0,
 590     17,  0,  0,  7,  0,  3,  0,  2,  0,  0,  1}
 600 m_vset(75,v)       /* Vocal ( Brass )
 610 /*
 620 /* AF  OM  WF  SY  SP PMD AMD PMS AMS PAN
 630 v={56, 15,  2,  0,200, 40,  0,  0,  0,  3,  0,
 640 /* AR  DR  SR  RR  SL  OL  KS  ML DT1 DT2 AME
 650     18, 16,  1,  2,  2, 28,  0,  5,  3,  0,  0,
 660     20, 31,  1,  2,  0, 28,  0,  3,  6,  0,  0,
 670     28, 31,  1,  2,  0, 28,  0,  1,  3,  0,  0,
 680     31, 31,  1,  7,  0,  4,  0,  1,  0,  0,  0}
 690 m_vset(76,v)       /* Distortion Guitar
 700 /*
 710 /* AF  OM  WF  SY  SP PMD AMD PMS AMS PAN
 720 v={60, 15,  2,  0,200, 40,  0,  0,  0,  3,  0,
 730 /* AR  DR  SR  RR  SL  OL  KS  ML DT1 DT2 AME
 740     10, 31,  1,  3,  0, 34,  2,  4,  3,  0,  0,
 750     12, 12,  0,  6,  1,  0,  1,  8,  3,  0,  0,
 760     14, 31,  1,  2,  0, 28,  2,  4,  7,  0,  0,
 770     10,  5,  0,  7,  1,  0,  1,  8,  7,  0,  0}
 780 m_vset(77,v)       /* Chorus
 790 /*
 800 /* AF  OM  WF  SY  SP PMD AMD PMS AMS PAN
 810 v={61, 15,  2,  0,200, 40,  0,  0,  0,  3,  0,
 820 /* AR  DR  SR  RR  SL  OL  KS  ML DT1 DT2 AME
 830     24, 12,  2,  5,  1, 25,  0,  8,  3,  0,  0,
 840     29, 31,  0,  8,  0,  8,  0,  8,  7,  0,  0,
 850     28, 31,  0,  7,  0,  8,  0,  8,  7,  0,  0,
 860     27, 31,  0,  7,  0,  6,  0,  4,  7,  0,  0}
 870 m_vset(78,v)       /* Synthe Brass
 880 /*
 890 /* AF  OM  WF  SY  SP PMD AMD PMS AMS PAN
 900 v={58, 15,  2,  0,200, 40,  0,  0,  0,  3,  0,
 910 /* AR  DR  SR  RR  SL  OL  KS  ML DT1 DT2 AME
 920     31, 23,  7,  5,  3, 38,  0,  3,  3,  0,  0,
 930     31, 23,  7,  6,  8,  5,  0,  3,  3,  0,  0,
 940     31,  0,  7,  6,  0, 53,  0,  6,  3,  0,  0,
```

▶今年は私も受験生となりまして，親にX68000をしまうように言われました。しかたないので本棚の下に隠して使っています。こんなときX68000PROでよかったと思います。
渡辺　一史(18)愛知県

```
 950     24,  0,  9,  6,  0,  0,  0,  1,  0,  0,  0}
 960 m_vset(79,v)       /* A Guitar
 970 /*
 980 /* AF  OM  WF  SY  SP PMD AMD PMS AMS PAN
 990 v={59, 15,  2,  0,200, 40,  0,  0,  0,  3,  0,
1000 /* AR  DR  SR  RR  SL  OL  KS  ML DT1 DT2 AME
1010     31, 20, 20,  5,  2,  0,  0, 15,  0,  3,  0,
1020     31, 20, 12,  5,  2, 35,  0,  8,  0,  2,  0,
1030     31, 20, 13,  5,  3, 33,  0,  7,  3,  1,  0,
1040     31, 20, 16, 10,  2,  0,  0,  2,  3,  0,  0}
1050 m_vset(80,v)       /* Cowbell
1060 /*
1070 /* AF  OM  WF  SY  SP PMD AMD PMS AMS PAN
1080 v={60, 15,  2,  0,200, 40,  0,  0,  0,  3,  0,
1090 /* AR  DR  SR  RR  SL  OL  KS  ML DT1 DT2 AME
1100     31,  2,  1,  0,  2, 28,  0,  3,  7,  0,  0,
1110     31, 12,  3,  8,  2,  0,  0,  2,  7,  0,  0,
1120     20,  4,  2,  1,  2, 30,  0,  5,  3,  0,  0,
1130     18, 12,  3,  8,  2,  0,  0,  1,  3,  0,  0}
1140 m_vset(81,v)       /* Synthe
1150 /*
1160 /* AF  OM  WF  SY  SP PMD AMD PMS AMS PAN
1170 v={60, 15,  2,  0,200, 40,  0,  0,  0,  3,  0,
1180 /* AR  DR  SR  RR  SL  OL  KS  ML DT1 DT2 AME
1190     31,  0,  0,  0,  0, 32,  0,  8,  7,  0,  0,
1200     18, 14,  8,  8,  3,  0,  0,  8,  7,  0,  0,
1210     31,  0,  0,  0,  0, 25,  0,  4,  3,  0,  0,
1220     31, 14,  8,  8,  3,  0,  0,  4,  3,  0,  0}
1230 m_vset(82,v)       /* E.Piano
1240 /*
1250 /* AF  OM  WF  SY  SP PMD AMD PMS AMS PAN
1260 v={61, 15,  2,  0,200, 40,  0,  0,  0,  3,  0,
1270 /* AR  DR  SR  RR  SL  OL  KS  ML DT1 DT2 AME
1280     15,  9,  0,  6,  5, 35,  0,  1,  3,  0,  0,
1290     15,  5,  0,  7,  2,  7,  0,  2,  7,  0,  0,
1300     16,  6,  0,  7,  2,  7,  0,  1,  7,  0,  0,
1310     17,  7,  0,  7,  2,  7,  0,  1,  7,  0,  0}
1320 m_vset(83,v)       /* Horn
1330 /*
1340 /* AF  OM  WF  SY  SP PMD AMD PMS AMS PAN
1350 v={22, 15,  2,  0,200, 40,  0,  0,  0,  3,  0,
1360 /* AR  DR  SR  RR  SL  OL  KS  ML DT1 DT2 AME
1370     31, 10,  7,  6,  1, 20,  0, 10,  7,  0,  0,
1380     31, 10, 10,  7, 10,  6,  0,  4,  7,  0,  0,
1390     31, 11,  7,  7, 10,  9,  0, 12,  3,  0,  0,
1400     31, 12,  8,  5,  6,  0,  0,  1,  3,  0,  0}
1410 m_vset(84,v)       /* Glass Bell
1420 /*
1430 /*  P r e p a r a t i o n s
1440 /*
1450 m_init()
1460 for i=1 to 8 : m_alloc(i,7000) : m_assign(i,i) : next
1470 /*
1480 str a(34)[256]
1490 str a1[70],a2[70],a3[70],a4[70],a5[70],a6[70],a7[70]
1500 str a8[70],a9[70],a10[70],a11[70],a12[70],a13[70],a14[70]
1510 str vc[20],dg[20],cg[20],br[20],ch[20],hn[20]
1520 /*
1530 vc="@75     q8 L8  o5" : dg="@76 v11 q8 L8
1540 cg="@79 v11 q8 L16 o5" : br="@78 v12 q8 L16 o3
1550 ch="@77 v11 q8 L1  o2" : hn="@83 v12 q8 L4  o4
1560 /*
1570 /*  R h y t h m s
1580 /*
1590 a1="@73<y2,10g4y2,22r>@71p1{cy2,10c}p2cy2,10ry2,22c.c16
1600 a2="y2,10c16y2,10r.y2,22c.y2,10r16cy2,10ry2,22c@72c@71
1610 a3="y2,10c4y2,22cp1{cy2,10c}p2cp1y2,10cp2y2,22c.@72c16@71
1620 a4="y2,10c16y2,10rc16y2,22r.y2,10c16cy2,10ry2,22c{cy2,22r}
1630 a5="@73<y2,10g4y2,22ry2,10r>@71cy2,10ry2,22cy2,10r
1640 a6="cy2,10ry2,22c.y2,10c.y2,10ry2,22c@72c@71
1650 a7="y2,10c4y2,22cy2,10rcy2,10ry2,22cy2,10r
1660 a8="cy2,10ry2,22c.y2,10c.{y2,42ry2,42r}y2,22c{cy2,22c}
1670 a9="@73<y2,10g16y2,10r.y2,22r4>@71cy2,10ry2,22c4
1680 a10="y2,10c.c16y2,22c.y2,10c16cy2,10ry2,22c{y2,22ry2,22r}
1690 a11="y2,10c16y2,10r.y2,22c.y2,10r16c{y2,42ry2,42r}y2,22c4
1700 a12="cy2,22r<@73y2,10g4y2,42r.y2,23r16y2,22r4>
1710 a13="cy2,10ry2,22c.y2,10c.{y2,42cy2,42r}y2,22c4
1720 a14="@73<y2,10g4y2,22r>@71p1{y2,10cc}p2cp1y2,10cp2y2,22cy2
,10r
1730 /*
1740 a( 0)="t141"+ch+"q8 o0 y3,3
1750 a( 1)="r r r a2<c+2 d c >g4.a2b-8& b-4<c2&y2,23c64&y2,22c1
6..&c8 v15 L8 o5
1760 a( 2)=a1+a2+a3+a4
1770 a( 3)=a1+a2+a3+"y2,10c4y2,22c.y2,10c16c{y2,42ry2,42r}y2,22
c{y2,22ry2,22r}
1780 a( 4)=a(2)
1790 a( 5)=a3+a2+a3+"y2,10c{y2,22rc}y2,22r16cy2,10c16cy2,10r@73
y2,22p1y152,9<c&y2,22c>
1800 a( 6)=a5+a6+a7+a8
1810 a( 7)=a5+a6+"y2,10c16y2,10r.y2,22c4cy2,10ry2,22c4 y2,10c16
y2,10r.y2,22c.L16y2,10ry2,22cy2,22ry2,10ry2,10ry2,22cry2,23ry2,2
2r L8
1820 a( 8)=a(2)
1830 a( 9)=a3+a2+a3+"y2,10c16y2,22r.y2,22c.y2,10r16{y3,2y2,41cy
2,41r}y3,3y2,10ry2,22c.y2,22r16
1840 a(10)=a1+a2+a3+a10
1850 a(11)=a3+a2+"y2,10c4y2,22cp1{y2,10cc}p2cp1y2,10cp2y2,22cy2
,10r cy2,22ry2,10r4L16y3,2y2,41ry2,41ry3,3y2,42ry2,42ry2,22ry3,1
y2,43rL8y2,43r  y3,3
1860 a(12)=a1+a2+a3+"y2,10c16y2,10r.y2,22c.{y2,10cy2,22c}y2,23r
y2,10r16y2,22c.y2,22r16
1870 a(13)=a1+a2+a3+"y2,10c16y2,10r.y2,22c.y2,10r16c{y2,42ry2,4
2r}y2,22c.y2,22r16
1880 a(14)=a1+a2+a3+"y2,10c{y2,22rc}y2,22r16cy2,10c16c{y2,42ry2
,42r}@73y2,22p1y152,9<c&y2,22c>
1890 a(15)=a5+a13+a5+a8
1900 a(16)=a5+a13+a9+"y2,10c16y2,10r.y2,22cL16y2,43ry2,43ry2,22
cy2,43ry2,43ry2,10ry2,22c8y2,23ry2,22r  L8
1910 a(17)=a(2)
1920 a(18)=a3+a2+a3+"y2,10c16y2,22r.y2,22c{y3,2y2,41ry2,41r}y3,
3y2,42cy2,42ry2,22c{y3,1y2,43ry3,3y2,22r}
1930 a(19)=a(10)
1940 a(20)=a3+a11+a14+"cy2,22ry2,10r4.{y2,42ry2,42r}y2,22r4
1950 a(21)="@73<y2,10g1> L16y2,22ry2,22r8y2,23ry2,10r8y3,2y2,41
ry2,41ry3,3y2,42ry2,42ry3,1y2,43r8y3,3y2,22r8.y2,22r L8
1960 a(22)=a(13)
1970 a(23)=a1+a2+a3+"y2,10cL16y2,22rcy2,22rcry3,2y2,41ry2,41cy3
,3y2,42rL8y2,42r@73y2,22p1y152,9<c&y2,22c>
1980 a(24)=a(15)
1990 a(25)=a5+a6+a9+"{y2,10cy2,10r}ry2,22c.L16y2,10ry2,22cy2,22
ry2,41ry2,42ry2,22c4  L8
2000 a(26)="@73<y2,10g1> |:5 r1 :| L16r2..y3,2y2,41ry2,41r ry2,
41ry2,41rry3,3y2,22rry2,10ry2,42r ry3,3y2,42ry2,42ry2,22ry2,22r4
 L8
2010 a(27)=a(10)
2020 a(28)=a3+a2+a14+"cy2,22rL16y2,10ry3,2y2,41rry3,3y2,42ry2,4
2rry3,1y2,43ry2,43ry3,3|:4y2,22r:| L8
2030 a(29)=a1+"L16y2,10cy2,10rry2,10ry2,22crry2,23rcry2,42ry3,1
y2,43ry3,3y2,22cy2,23rry2,22r L8
2040 a(30)=a(8)
2050 a(31)=a3+a11+"y2,10c4y2,22rp1{cy2,10c}p2cp1{y2,10cy2,42r}p
2y2,22c4 L16y2,10cy2,22ry2,22ry2,41ry2,22crry2,42ry2,42cy2,43ry2
,43rry2,22crry2,22r L8
2060 a(32)=a(10)
2070 a(33)=a3+"y2,10c16y2,10r.y2,22c.L16y2,10rcy3,2y2,41ry2,41r
y3,3y2,42ry2,22cy2,42ry3,1y2,43ry2,43r y3,3L8"+a14+a12
2080 a(34)=a14+a12+a14+"cy2,22r<@73y2,10g4.{y2,10ry2,23r}y2,22r
4 y2,10g1>
2090 m_trns(1)
2100 /*
2110 /*  B a s s
2120 /*
2130 a( 0)=ch+"y49,00
2140 a( 1)="f& f e e f& f& f& f  @74 q7 v12 L8 o3
2150 a( 2)="fr>f.<f16{r>a}<c4{dc} >{ff}r<fc16ff16e-dc >fr<c.fe-
16f4{fc} >fr<c.ff16e-dc
2160 a( 3)="frc.f>{ab-}<c.{dc} >f16f.<c.ff16gfc >fr<c.f>f16ab-<
c >fr<c.e-e-16dc4
2170 a( 4)="fr>f.<f16{r>a}<c4{dc} >fr<fc16f>a16b-<c4 >fr<fc16f>
{ab-}<c.{dc} >fr<f.gg16fc{fc}
2180 a( 5)=">fr<fc16f>{ab-}<c.{dc} >{ff}r<fc16f>a16<c4. >fr<fc1
6f>a16b-<c4 >f16f.<fc16ff16gfc>
2190 a( 6)="|:3 a<a>ab-4<b->b-<c& |1 cccc16cc16dc>b-< :||2 cccc
16cc16>g<c>b-< :|
2200 a( 7)="|3 cccc16<cc16>gcr e-4e->b-16<e->b-16<e-dd- c4c4{cc
<cc}4>cr
2210 a( 8)="f4.c16f>a16<c4{dc} >f4.<d16ff16cdc e4.c16ee16>egg+
a4..ar16a<c>a
2220 a( 9)="b-4..<b-b-16fc>b- a4..<aa16fc>a g+4..<g+>g+16g+<g+>
g+ g4<d-4c4>{g&g-&f&e&d+&d&c+&c}4
2230 a(10)="<f4c.ff16>a<c4 >f4<f>f16aa16<dcf e4.>a16<ee16edd- >
a4.{ga}<d-4.>{a<d-}
2240 a(11)="d4.r16dd16dd-c >b4<b.>bb16def g<g>ra4a4b-& b-r<c2.
2250 a(12)=">fr<f>f16<f>a16b-<c{fc} |: >fr<f>f16<f>{ab-}<c.{fc}
:| >fr<fc16ff16e-dc
2260 a(13)="frfd16f>{ab-}<c.{fc} {ff>fr}4<fc16f>a16{b-<c&}c4 f>
f<fc16f>{ab-}<c.{fc} {ff>fr}4<fc16ff16gfc
2270 a(14)=">fr<fc16f>{ab-}<c.{fc} >{ff<cr}4fc16f>a16b-<c4 >f4<
fc16f>a16<c4{fc} >f4<fc16ff16e-dc>
2280 a(15)=a(6)
2290 a(16)="|3 cc<c>c16cc16<c>cd e-4e-4>e-<e-dd- c4c4ccdc
2300 a(17)="f4..fr16fcf >fa<c.ff16dfc e4>e.<er16e>gg+ a4..ar16a
<c>a
2310 a(18)="b-4.f16<b-b-16fd>b- a4a.ar16a<a>a g+4..g+r16g+4g& g
ab-b<cdc>f
2320 a(19)="<f4f>f16a.<c4{fc} >f4<c.g.fcd e4.>a16<ee16edd- >a4.
{ga}<d-4.>{a<d-}
2330 a(20)="d4.r16dd16dd-c >b4<b.>bb16def g<g>ra4a4b-& b-r<c2.
2340 a(21)=">b-1& b-2.{b-&a&g+&g&f+&f&e&e-}4
2350 a(22)="<f4c.f16{r>a}<c4{fc} {ff>fr}4<fc16ff16gfc f>f<fc16f
>{ab-}<c.{fc} >fr<f.gg16fc{fc}
2360 a(23)=">fr<fc16f>{ab-}<c.{fc} >f16f.<f.ff16e-fc >f4<fc16f>
{ab-}<c.{fc} >f4<fc16cc16fc4>
2370 a(24)=a(6)
2380 a(25)="|3 cc<c>c16cc16<c>cd e-4e-.>b-16<{e-r}e-dd- c4c4c4c
8.&{d-&d&e-&e}16
2390 a(26)=ch+">f& f e a b- a a- g  @74 q7 v12 L8 o2
2400 a(27)="<f4..ff16>a<cf >f4<c.ff16cfd e4..<e>e16ede >a4.{ga}
<d-4.>{a<d-}
2410 a(28)="d4>a.<dd16dd-c >b4<b.>ba16b<b>b g<g>ga4a4b-& b-<b-c
2.
2420 a(29)="c+4c+.>a-16<{c+r}c+4c+ cc<c>c<c16>gg16{cc}>g<
2430 a(30)="f4c.f>a16<c4{fc} f4c.fc16gfc e4>e.<e.e>gg+ a4<e.a.e
>a4
2440 a(31)="b-4b-.<b-b-16fd>b- a4a.<aa16fc>a g+4g+.g+.g+fg& gab
-b<cdc4
2450 a(32)="f4.c16ff16>a<c{fc} >f4<c.ff16cfd e4>e.<ee16e>gg+ a4
.{ga}<d-4.>{a<d-}
2460 a(33)="d4>a.<dr16dd-c >b4&b16rb.b4. g<g>ga4<a&{a&a-&g&g-&f
&e&d+&d}>b-& b-4<c4.c>b-a
2470 a(34)="g<g4>a4a4b-& b-<b-c4.>g<c4 >g<g4>a4a4b-& b-b-<c2. >
f1&f1
2480 m_trns(2)
2490 /*
2500 /*  M e l o d y
2510 /*
2520 a( 0)=vc+"v14 y50,00
2530 a( 1)="a2.g4 b-4a4g&ffg& g2r4e4 a4b-4a4gg& gff2.& f2r2 ab-
4a4b-4<c& c&>ff4.fg{f&g&}
2540 a( 2)="f1 r1 r1 "+dg+"o3 r4.q5{ff}q7<f4f4
2550 a( 3)="f.&{f&e&d+&d&c+&c}16r2. r1 r1 r2"+br+"ffr8f4"+a(0)
2560 a( 4)="c{cr}cc4{cr}cc cd4c4.r4 >b-{b-r}b-b-4{b-r}b-b- ab-4
<c4.r4
2570 a( 5)="c{cr}cc4{cr}cc e-d4c4.r4 f{fr}ffff4f& f4r2.
2580 a( 6)="f4df4f4g& gededc4r f4df4f4g& g{cr}cc{dr}crc
2590 a( 7)="f4df4f4g& grcc{dr}c4r b-2{ar}g4f r<ccc>agfr
2600 a( 8)="a2.g4 b-4a4g&ffg& g2r4e4 a4g4f4ee&
2610 a( 9)="eff4.r4g f&ccc4.rc d-4fa-4b-4<c& c>b-<c>{b-<c}>b-4r
4
2620 a(10)="a2.g4 b-4a4g&ffg& g2r4e4 a4b-4&a4gg&
2630 a(11)="gff2.& f2r2 ab-4a4b-4<c& c>ff4.fg{f&g&}
2640 a(12)="f1 r1 r1"+br+"r2ffr8f4"+a(0)
2650 for i=13 to 20 : a(i)=a(i-9) : next
2660 a(21)="f1& f4r4 "+dg+"v14o4 b-2
2670 a(22)="L16{a&b-&a&b-&a&}4a8fre-&f&f4e-c e-e-c>b-aa<c>fe-e-
c>b-aa<c>a ab-<c>ab-age-f&g&a-f&f4 <{d&e-&}e-8{d&e-&}e-8c8&c4&{c
&>b&a+&a&g+&g&f+&f}4
2680 a(23)="<a&b-&a&b-&a8agf4&{e&e-&}df8& f4d8.fdc>b-gfga<c fga
<c8.{>fa<c}8fgfdcdc>a b-ga<c8c{e-&e&f&}8f4&{f&e&d+&d&c+&c}4"+vc
2690 a(24)=a(6) : a(25)=a(7)
2700 a(26)=a(8)+a(9)
```

▶新しいFloat2.Xを入手した。おお，速いぞ。SX-WINDOWもよくなったし，あとはXCをなんとかしてほしいな。

松井　和宏(22)東京都

```
 2710 a(27)=a(10) : a(28)=a(11)
 2720 a(29)="f2.r4 r1
 2730 for i=30 to 32 : a(i)=a(i-22) : next
 2740 a(33)="gff2.& f2r2 |:3 ab-4a4b-4<c& |1 c>fff4fdc< :|
 2750 a(34)="|2 crcc{dr}c4r :||3 c>ff4rfg{f&g&} f1& f2r2
 2760 m_trns(3)
 2770 /*
 2780 a( 0)=vc+"v11 y51,24 r8
 2790 a( 2)="f1 r1 r2.. "+dg+"o2 r4.q5{ff}q7<f4f4
 2800 a( 3)="f.&{f&e&d+&d&c+&c}16r2. r1 r1 r2"+br+">ffr8f4"+a(0)
 2810 a(12)="f2.r4 r1 r2.."+br+"r2>ffr8f4"+a(0)
 2820 a(21)="f1& f4r4 "+dg+"o4 b-2
 2830 m_trns(4)
 2840 /*
 2850 /* S u b
 2860 /*
 2870 a1="v12 r1 r8g+32&a.grfrgrdf8.dr r2..g+32&a.& a8arb-rbr<cr
>fg+8.ar
 2880 a2="o4|:6 r1 :| @84v10p2L8 d-4fa-4b-4g& g<d4"+br+"v13p2
 2890 /*
 2900 a( 0)=br+"y52,00 L1
 2910 a( 1)="r r r r r r L16r2..v15f&v11f& v7f&v4f&v5f&v6f&v7f&v
8f&v9f&v10f&v11f&v12f&v13f&v14f&v15f4
 2920 for i=2 to 5 : a(i)="|:"+cg+"p1ccccr2rrp2cc ccccr4@80v12o4
frrfr4 :|" : next
 2930 a( 6)=br+a1
 2940 a( 7)="r1 r1 <c&v11c&v10c&v9c&v8c&v7c&v6c8.&v7c&v8c&v9c&v1
0c&v11c&v12c8 q6L8rccrq8crc.r16
 2950 a( 8)="r2"+cg+"p1crrcr4 r2p2ccrcr4 r2p1crrcr4 r2p2c+c+rc+r
4
 2960 a( 9)="r2p1drrdr4 r2p2ccrcr4 r2p1c+rrc+r4 r4."+br+"<p2c&v1
0c&v7c&v8c&v9c&v10c&v11c&v12c&v13c&v14c
 2970 a(10)=a(8)
 2980 a(11)="r2p1>arrar4 r2p2bbrbr4 "+br+"v11>L8g4r<d4>b-<df& f{
gr}<c2&cr
 2990 for i=12 to 25 : a(i)=a(i-9) : next
 3000 a(21)="r4."+cg+"v14{e&f&}f&f2& f1
 3010 a(26)=a2+"f4frd8r8
 3020 a(27)=a(10) : a(28)=a(11)
 3030 a(29)="r2"+cg+"p1c+rrc+r8"+br+"b32&<c.& c8>brb-rarb-r<c>a8
cdf
 3040 for i=30 to 32 : a(i)=a(i-22) : next
 3050 a(33)="r2p1>arrar4 r2p2bbrbr4 r2.."+br+"<c8& c4.>arb-r8ar8
fg
 3060 a(34)="r1 rrdrr4drrfrrg8 L8rb-4<d4>b-<df& f{gr}<c2&cr"+ch+
"v8>f& f& {f&v7f&v6f&v5f&v4f&v3f&v2f&v1f}
 3070 m_trns(5)
 3080 /*
 3090 a( 0)=br+"y53,24 L1>
 3100 for i=2 to 5 : a(i)="|:"+cg+"p1ffffr2rrp2ff ffffr4@80v12o4
frrfr4 :|" : next
 3110 a( 6)=br+">"+a1
 3120 a( 7)="r1 r1 <f&v11f&v10f&v9f&v8f&v7f&v6f8.&v7f&v8f&v9f&v1
0f&v11f&v12f8 q6L8rggrq8grg.r16
 3130 a( 8)="r2"+cg+"p1frrfr4 r2p2ffrfr4 r2p1errer4 r2p2eerer4
 3140 a( 9)="r2p1frrfr4 r2p2ffrfr4 r2p1frrfr4 r4."+br+"p2g&v10g&
v7g&v8g&v9g&v10g&v11g&v12g&v13g&v14g
 3150 a(10)=a(8)
 3160 a(11)="r2p1drrdr4 r2p2ddrdr4 "+br+"v11>>L8g4r<d4>b-<df&f{g
r}<c2&cr
 3170 for i=12 to 25 : a(i)=a(i-9) : next
 3180 a(21)="r4."+cg+"v14{a&b-&}b-&b-2& b-1
 3190 a(26)=a2+"<f4frd8r8
 3200 a(27)=a(10) : a(28)=a(11)
 3210 a(29)="r2"+cg+"p1frrfr8"+br+">b32&<c.& c8>brb-rarb-r<c>a8c
df
 3220 for i=30 to 32 : a(i)=a(i-22) : next
 3230 a(33)="r2p1drrdr4 r2p2ddrdr4 r2.."+br+"c8& c4.>arb-r8ar8fg
 3240 m_trns(6)
 3250 /*
 3260 /* B a c k i n g
 3270 /*
 3280 a( 0)=ch+"y54,00 q8
 3290 a( 1)="a& a g a a& a a& a
 3300 for i=2 to 5 : a(i)="@81 v8 L16 o4 |: p1cdf<cr2r8>p2fd fab
-<cp3e-dc>fp1<fe-dcp2e-dc>f :|" : next
 3310 a( 6)=dg+"o2|:3 a4.b-2<c& c4>v15c&{v14c&v13c&v12c&v11c&}4c
4. :|
 3320 a( 7)=" r1 r1
 3330 a( 8)="r2..@82v12L8o3c& c1 r4.c&c2 r4.c+&c+2
 3340 a( 9)="r4"+hn+"df<c8.&>b32&b-32& a1 @84v10o4p2 d-f8a-b-g8&
 g8<df8&f2
 3350 a(10)=a(8)
 3360 a(11)="r4"+hn+"fa<d8.&c+32&c32& >b1 r1 r1
 3370 for i=12 to 25 : a(i)=a(i-9) : next
 3380 a(21)=cg+"v14o3a32&b-8..&b-2.& b-1
 3390 a(26)=ch+"<c& c c c+ d c >b-& b- L8
 3400 a(27)=a(10) : a(28)=a(11) : a(29)="r1 r1
 3410 for i=30 to 32 : a(i)=a(i-22) : next
 3420 a(33)="r4"+hn+"fa<d8.&c+32&c32& >b1 r1 <rv10d2.
 3430 a(34)="r1 r1"+ch+"v8>g4.a2b-8& b-4<c2. a& a& {v7a&v6a&v5a&
v4a&v3a&v2a&v1a&v0a}
 3440 m_trns(7)
 3450 /*
 3460 a( 0)=ch+"y55,20 q8 y15,0
 3470 a( 1)="c& c c c+ d c d4.c2d8& d<
 3480 a( 2)="v7f e- d e-" : a(3)=a(2) : a(4)=a(2)
 3490 a( 5)="f e- d c+
 3500 a( 6)="|:3 c4.d2d8& d :|
 3510 a( 7)="e-1 d1
 3520 a( 8)="f& f e& e
 3530 a( 9)="d f c+ d
 3540 a(10)="f& f e c+
 3550 a(11)="d& d d4.d2d8& d
 3560 for i=12 to 25 : a(i)=a(i-9) : next
 3570 a(21)=cg+"v14o4e32&f8..&f2.& f1"+ch+"<
 3580 a(26)=ch+"a& a g e f& f& f& f v7<
 3590 a(27)=a(10) : a(28)=a(11) : a(29)="c+ d
 3600 for i=30 to 33 : a(i)=a(i-22) : next
 3610 a(34)="|: d4.d2d8& d :| v8f1& f1& {v7f&v6f&v5f&v4f&v3f&v2f
&v1f&v0f}
 3620 m_trns(8)
 3630 /*
 3640 /* P l a y i n g
 3650 /*
 3660 m_play()               /* ブルーウォーター -Miho Morikawa-
 3670 end                    /*
 3680 func m_trns(x)         /* Programmed by Shinya Kanoh
 3690   for i=0 to 34        /* Sound       by Yoshinori Kohara
 3700   m_trk(x,a(i))        /*
 3710   next                 /* Presented  by Midy House
 3720 endfunc                /*
```

リスト3 ナディア用コンフィグファイル

```
10=¥SAMPLING¥KICK.PCM
22=¥SAMPLING¥SD2V13.PCM
23=¥SAMPLING¥SD2V10.PCM
41=¥SAMPLING¥TOM2_2.PCM
42=¥SAMPLING¥TOM2_3.PCM
43=¥SAMPLING¥TOM2_4.PCM
```

日本音楽著作権協会(出)許諾第9170217-101号

リスト4 POWER HOLL

```
 10 /*
 20 /*  P O W E R  H O L L( 長州 力  テーマ曲 )
 30 /*
 40 key 19,"m_tempo(150)@M":key 20,"m_tempo(200)@M
 50 m_init():for z=1 to 8:m_alloc(z,5000):m_assign(z,z):next
 60 str y1="y2,5",y2="y2,14",y3="y2,16",y4="y2,15"
 70 str r="r4",rr="r8",R="r16":char v(4,10)
 80 str a[256],b[256],c[256],d[256],e[256],f[256],g[256]
 90 str h[256],i[256]
100 /* リードシンセ18(電波新聞社 音色ライブラリーvol.2より)
110 v={44,15,0,0,0,0,0,0,0,3,0,
120     31,0,0,0,0,22,0,2,7,0,0,
130     18,10,0,6,1,0,0,8,7,0,0,
140     31,0,0,0,0,23,0,4,3,0,0,
150     18,10,0,6,1,0,0,4,3,0,0}
160 m_vset(70,v)
170 /* クラビネット1( " )
180 v={59,15,0,0,0,0,0,0,0,3,0,
190     31,5,4,7,15,40,3,11,0,0,0,
200     31,5,4,7,1,30,0,7,0,0,0,
210     31,5,4,7,1,30,0,0,0,0,0,
220     31,5,2,8,1,0,3,1,0,0,0}
230 m_vset(71,v)
240 /* コーラス(電波新聞社 B A S I C  M a g . 1991.4より)
250 v={60,15,0,0,0,0,0,0,0,3,0,
260     31,15,7,3,3,17,0,8,3,0,0,
270     14,25,0,6,0,0,0,4,3,0,0,
280     31,15,7,3,3,17,0,8,7,0,0,
290     14,25,0,6,0,0,0,4,7,0,0}
300 m_vset(72,v)
310 /*
320 /*
330 /*******太鼓 パターン(ch.1)*******
340 a="t150
350 b="y3,3"+r+y1+R+"y3,3"+y1+R+"y3,3"+y1+R+"y3,3"+y1+R
360 c="y3,3|:14"+r+y2+r+":|":d="y3,3|:6"+r+y2+r+":|"
370 e="y3,3|:2"+r+y2+r+":|":f="y3,3|:4"+r+y2+r+":|"
380 g=r+y2+r+r+"y3,3"+y2+R+"y3,3"+y2+R+y2+R+"y3,3"+y2+R
390 h="y3,2"+y3+rr+"y3,1"+y3+rr+"y3,2"+y4+rr
400 i="y3,1"+y4+rr+"y3,3"+y2+rr+y1+"q4 v13 o4 @70 d4"+y1+"q8 e
4"
410 m_trk(1,a+b+"[$]"+c+g+d+g):m_trk(1,c+g+c+d+e+g+"[d.s.]")
420 m_trk(1,c+f+h+i)
430 /*
440 /*
450 /*******16拍子 で後ろで鳴ってるやつ(ch.2,3)*******
460 a="t150 L16 o2 q7 v12 @71 y50,40 r2":b="|:32aaaa:|
470 c="<|:2ccccccccddddddddcccccccceeeeeeee:|>
480 d="<|:4ffffffffggggggggffffffffgggggggg:|>
490 e="|:6aaaa:|":f="r16.. v13 o3 q4 p3 @70 b4 q8<c4
500 m_trk(2,a+"[$]p1"+b+c+b+c+d+"[d.s.]"):m_trk(2,b+e+"r64"+f)
510 m_trk(3,a+"r64[$]<p2"+b+c+b+c+d+"[d.s.]"):m_trk(3,b+e+f)
520 /*
530 /*
540 /*******主旋律(ch.4,5)*******
550 a="t150 L8 o3 v15 @70 p3 r2
560 b="t150 L8 o3 v12 @70 p3 y52,40 r2r8
570 c="L8|:4a4&<a4&e&g&g4&g&e&d&e&c&d&c&>b&:|
580 d="<|:2c2&d2&c2&e2&:|>
590 e="<|:6f2&g2&:|>
600 f="<|:2f4&g&a&g4&e4&:|>
610 g="v15 a4&<a4&e&g&g4&g2 q4 r8g4 q8 a4
620 h="v14 a4&<a4&e&g&g4&g2 q4 g4 q8 a4
630 m_trk(4,a+"[$]"+c+d+c+d+e+f+"[d.s.]"):m_trk(4,c+g)
640 m_trk(5,b+"[$]"+c+d+c+d+e+f+"[d.s.]"):m_trk(5,c+h)
650 /*
660 /*
670 /*******コーラス(ch.6)*******
680 a="t150 L8 o2 v5 @72 p2 r2
690 b="|:16a2&:|
700 c="<v13|:2c2&d2&c2&e2&:|v5>
710 d="<v10|:8f4&g&a&g4&e4&v12:|v5>
720 e="|:9r1:|r2r8 @70 q4 p3 v13 o4 d4 q4 e4
730 m_trk(6,a+"[$]"+b+c+b+c+d+"[d.s.]"):m_trk(6,e)
740 /*
750 /*
```

```
760 /*******副旋律(ch.7,8)*******
770 a="t150 L8 o3 v15 @70 p2 r2
780 b="t150 L8 o3 v15 @70 p1 y55,40 r2r64
790 c="|:16r2:|":d="|:2c4&d&e&d2&c4&d&ee2&:|
800 e="r1r1r1r1|:4f4&g&a&g4&e4&:|
810 f="|:8r1:|>r2a4&<a4&g2 q4 v15 r8g4 q8 a4
820 g="|:8r1:|>r2a4&<a4&g2 q4 v15 r16..g4 q8 a4
830 m_trk(7,a+"[$]"+c+d+c+d+e+"[d.s.]"):m_trk(7,f)
840 m_trk(8,b+"[$]"+c+d+c+d+e+"[d.s.]"):m_trk(8,g)
850 m_play()
```


リスト5　見知らぬ国のトリッパー

```
10 '
20 ' ﾏﾎｳ ﾉ ﾖｳｾｲ ﾍﾟﾙｼｬ  ﾖﾘ
30 '
40 '  「ﾐｼﾗﾇ ｸﾆ ﾉ ﾄﾘｯﾊﾟｰ」
50 '
60 '                By kunkun
70 '
80 '            1991/03/??
90 '
100 INIT:GOSUB 3060
110 DEFSTR a-h,r
120  r="r:"
130   GOTO 190
140 '
150 LABEL "!"
160  PLAY a;:PLAY b;:PLAY c;:PLAY d;:PLAY
 e;:PLAY f;:PLAY g;:PLAY h
170 RETURN
180 '
190 TEMPO0
200 PLAY STRING$(8,"O4L1I1:")+"t140"
210 '
220 a="V15>E4.E8D2<:"
230 b=r :c=r :e=r :g=r :h=r
240 d="V12R8F#8A4.E8F#8A8&:"
250 "!"
260 d="A8F#8A4.E8F#8A8&:"
270 "!"
280 a=">C#4.C#8<B2:"
290 d="A8D8F#4.C#8D8F#8&:"
300 "!"
310 d="F#8D8F#4.C#8D8F#8:"
320 "!"
330 a="A4.A8G4.G8:"
340 d="R8<B8>D4.<A8>D4:"
350 "!"
360 "!" ' 1ｼｮｳｾﾂ ｸﾘｶｴｼ
370 a="F#4.F#8E4.A8:"
380 d="R8<A8>D4.<B8>C#4:"
390 "!"
400 "!" ' 1ｼｮｳｾﾂ ｸﾘｶｴｼ
410 a="R4<A4>D4G4:"
420 b="R2A4R4:"
430 d="V11E8F#8A8>E8D8<E8F#8A8:"
440 e="V12<D4.D8D2:"
450 "!"
460 a="<B4>F#4F#4E4:"
470 b="G4R2R4:"
480 d=">C#8<D8F#8>C#8<B8C#8D8F#8:"
490 e="<B4.B8B2:"
500 "!"
510 a="E8D8C#8D8&D2&:"
520 b=r
530 d="A8<B8>D8A8G8<A8B8>D8:"
540 e="G4.G8G2:"
550 "!"
560 a="D2.<A4>:"
570 d="E8<F#8A8>E8D8<E8F#8A8>:"
580 e="F#4.F#8F#2:"
590 "!"
600 a="<G4>D4D8E4D8&:"
610 b="<B4R4R8R4R8>:"
620 d="B8G8G8B8>G8<B8>D8G8:"
630 e="E4.E8E2:"
640 "!"
650 a="D2.F#4:"
660 b=r
670 d="D8<A8A8>D8F#8<A8>D8F#8<:"
680 e="F#4.F#8F#2:"
690 "!"
700 a="G4F#4E4D4:"
710 d="B8D8D8G8B8D8F#8B8:"
720 e="G4.G8G#2:"
730 "!"
740 a="F#4.E8&E2:"
750 d=">C#8<E8E8A8>A8G8F#8E8:"
760 e="A4.A8A4B8>C#8:"
770 "!"
780 a="R4<A4>A4G4:"
790 d="E8<F#8A8>E8D8<E8F#8A8:"
800 e="D4.D8D2:"
810 "!"
820 a="G4F#4F#4E4:"
830 d=">C#8<D8F#8>C#8<B8C#8D8F#8:"
840 e="<B4.B8B2:"
850 "!"
860 a="E4F#8D8&D2&:"
870 d="A8<B8>D8A8G8<A8B8>D8:"
880 e="G4.G8G2:"
890 "!"
900 a="D2.<A8A8>:"
910 d="E8<F#8A8>E8D8<E8F#8A8>:"
920 e="F#4.F#8F#2:"
930 "!"
940 a="<B4>D4D8D8D8<B8>:"
950 d="G8<B8>D8G8A8D8F#8A8:"
960 e="E4.E8F#2:"
970 "!"
980 a="G2F#4E8D8&:"
990 d="G8D8G8B8A8E8G8A8:"
1000 e="G4.G8A2:"
1010 "!"
1020 a="D2R8A8A8A8:"
1030 b="v15R2R8>C#8C#8D8<:"
1040 c="v15R2R8>E8E8F#8<:"
1050 d="F#2R8<A8A8>D8:"
1060 e="D4.D8R8D8D8D8>>:"
1070 "!"
1080 a="R2R8Q5D8Q7Q5E8Q7F#8&:"
1090 b=r :c=r :d=r
1100 e="R2R8Q5<<B8Q7Q5A8Q7G8&>>:"
1110 "!"
1120 a="F#4F#4E8E8A8E8&:"
1130 d="R4<B4>C#2:"
1140 e="R4D4E2:"
1150 f="R4F#4R2:"
1160 g="<<G4.G8F#4.F#8:"
1170 h="V11R2C#8E8A8E8"
1180 "!"
1190 a="E8D4.R8D8E8F#8&:"
1200 d="R8D8D8F#8A8D8D8R8:"
1210 e=r : f=r :h=r
1220 g="B4.B8R8B8A8G8&:"
1230 "!"
1240 a="F#4F#4E8E8A8E8&:"
1250 d="R4<B4A2>:"
1260 e="R4D4C#2:"
1270 f="R4F#4E2:"
1280 g="G4.G8F#4.F#8:"
1290 h="R2C#8E8A8E8"
1300 "!"
1310 a="E8D4.R8Q5D8Q7Q5E8Q7F8&:"
1320 d="R8D8D8F#8A8D8D8<G8&:"
1330 e="R2R4R8<B8&:"
1340 f="R2R4R8D8&:"
1350 g="B4.B8R8Q5B8Q7Q5A8Q7G8&:"
1360 h=r
1370 "!"
1380 a="F4F4E8E8G8E8&:"
1390 d="G4G4G4>R4:"
1400 e="B4B4>C4R4:"
1410 f="D4D4E4R4:"
1420 g="G4.G8>C4.C8:"
1430 "!"
1440 a="E8C8&C2R8D8&:"
1450 d="R8<A4>R3...:"
1460 e="R8C4R3...:"
1470 f="R8F4R3...:"
1480 g="<F4.F8>D8D4<G8&:"
1490 "!"
1500 a="D8F8F8D8F8G4A8&:"
1510 d="R2R4R8C8&:"
1520 e="R2R4R8E8&:"
1530 f=r
1540 g="G4.G8>C8C4<F8&:"
1550 "!"
1560 a="A2.R8D8&:"
1570 b="R2.R8G8&Q7:"
1580 c="R2.R8B8&Q7:"
1590 d="C8C8Q5D8Q7R8C8Q5D8Q7R8<G8&>:"
1600 e="E8E8Q5F8Q7R8E8Q5F8Q7R8E8&:"
1610 g="F8F8F4F8F4A8&>>:"
1620 h="R2R4R8G8&"
1630 "!"
1640 a="D8Q5D8Q7E4R8A8G8A8&:"
1650 b="G8Q5G8Q7A4R2:"
1660 c="B8Q5B8Q7>C#4R2<:"
1670 d="<G8Q5>E8Q7F#4R2:"
1680 e="E8Q5G8Q7A4R2:"
1690 f="<<A8Q5A8Q7A2.>:"
1700 g="G8Q5G8Q7Q5A8Q7R3...:"
1710 h=r
1720 "!"
1730 a="A:"
1740 b=r :c=r :g=r
1750 d="R4D4C8D4C8&:"
1760 e="R4F4E8F4E8&:"
1770 f="D4.D8D4.D8:"
1780 "!"
1790 a="R4R8<A8>A8G8F8A8&:"
1800 d="C8D8D8R3...:"
1810 e="E8F8F8R3...:"
1820 f="D4.D8D4.D8>:"
1830 "!"
1840 a="A8A#8A#2.:"
1850 d="R4G4D8E4D8&:"
1860 e="R4A#4F8G4F8&:"
1870 f="<<G4.G8G4.G8:"
1880 "!"
1890 a="R2R8G8F8G8&:"
1900 d="D8E8E4R2:"
1910 e="F8G8G4R2:"
1920 f="G4.G8G4.G8>:"
1930 "!"
1940 a="G:"
1950 d="R4C4D8E4D8&:"
1960 e="R4E4F8G4F8&:"
1970 f="C4.C8C4.C8:"
1980 "!"
1990 a="R4R8C8>C8<G8G8G8&:"
2000 d="D8E8E4R2:"
2010 e="F8G8G4R2:"
2020 f="C4.C8C4.C8:"
2030 "!"
2040 a="G8A8A2R8D8&:"
2050 b="R4R2R8G8&:"
2060 c="R4R2R8B8&:"
2070 d="R4D4C8D4E8&:"
2080 e="R4F4E8F4G8&:"
2090 f="<F4.F8F8F4A8:"
2100 "!"
2110 a="D8Q5D8Q7E4R8A8G8A8&:"
2120 b="G8Q5G8Q7A4R2:"
2130 c="B8Q5B8Q7>C#4R2<:"
2140 d="E8Q5E8Q7F#4R2:"
2150 e="G8Q5G8Q7A4R2:"
2160 f="A8Q5A8Q7A2.>:"
2170 "!"
2180 a="A:"
2190 b=r :c=r
2200 d="R4D4C8D4C8&:"
2210 e="R4F4E8F4E8&:"
2220 f="D4.D8D4.D8:"
2230 "!"
2240 a="R4R8D8A8G8F8A8&:"
2250 d="C8D8D8R3...:"
2260 e="E8F8F8R3...:"
2270 "!"
2280 a="A8A#8G2.:"
2290 d="R4E4D8E4D8&:"
2300 e="R4G4F8G4F8&:"
2310 f="<G4.G8G4.G8:"
2320 "!"
2330 a="R2G8A8A#8>C8&:"
2340 d="D8E8E4R2:"
2350 e="F8G8G4R2:"
2360 f="G4.G8G4.G8:"
2370 "!"
2380 a="C8<F8F2R8C8:"
2390 d=r :e=r
2400 f="Q4A4.A8A8A4A8>>:"
2410 g="Q4<C4.C8C8C4C8:"
2420 "!"
2430 a="D4F4A#4A4:"
2440 f="<<A#4.A#8A#8A#4A#8>:"
2450 g="D4.D8D8D4D8>:"
2460 "!"
2470 a="A4.G8&G4R8G8&:"
2480 b="R4.R2B8&:"
2490 c="R4.R2>D8&<:"
2500 d="v15R2>C8<B8A#8A8&:"
2510 f="C8C4C8C8<B8A#8A8&:"
2520 g="<E8E4E8R2>:"
2530 "!"
2540 a="G2A4G4:"
2550 b="B2R2:"
2560 c=">D2R2<:"
2570 d="A8R2..:"
2580 f="A2.R4>>:"
2590 g=r
2600 "!"
2610 a="F#4G8A8&A2:"
2620 b=r :c=r
2630 d="V15R4R8A8>A4G4<:"
2640 f="<D4.D8D2>:"
2650 g=">E8<F#8A8>E8V11D8<E8F#8B8:"
2660 "!"
2670 a=">G4F#4F#4E4<:"
2680 d=r
2690 f="<<B4.B8B2:"
2700 g=">C#8<D8F#8>C#8<B8C#8D8F#8:"
2710 "!"
2720 a=">E4F#8D8&D2&<:"
2730 f="G4.G8G2:"
2740 g="A8<B8>D8A8G8<A8B8>D8:"
2750 "!"
2760 a=">D2.<A8A8:"
2770 f="F#4.F#8F#2:"
2780 g="E8<F#8A8>E8D8<A8B8>D8:"
2790 "!"
2800 a="T135G2A2:"
2810 b="B2>D2<:"
2820 f="E4.E8F#2:"
2830 g="G8B8>D8<G8A8>C#8E8A8<:"
2840 "!"
2850 a="B2B2:"
2860 b=">D2D2<:"
2870 c=">G2A2<:"
2880 f="G4.G8A2>>:"
2890 g="B8>D8G8B8<A8>E8G8>C#8:"
2900 "!"
2910 a="T134>E4.E8D2<:"
2920 b=r :c=r :e=r :f=r
2930 d="R8F#8A4.E8F#8A8:"
2940 g="V15D2V11<<R2:"
2950 "!"
2960 a="T133>E4.E8D2<:"
2970 d="R8F#8A4.E8F#8A8:"
2980 g=r
2990 "!"
3000 a="T230R4A&A2&A1&A4:"
3010 b="R8R4>D2&D&D4:"
3020 d="D&D2&D1&D4:"
3030 e="R8F#2&F#1&F#:"
3040 "!"
3050 END
3060 MEM$(&HB190,36)=HEXCHR$("FA 00 31 4C
 33 51 25 2F 2F 00 9C 96 5D 8F 04 09 04 8
7 00 01 03 00 15 12 16 A5 80 80 80 80 00
DC 80 00 02 80")'PIANO
3070 RETURN
```

▶ついにC言語実習(会社で)が始まった。これでやっとXCがまともに使えるようになりそう。ちなみに会社のマシンはオムロンのワークステーションLUNAの予定。

阪長　俊之(24)京都府

戦えロボット君2（前編）

プロジェクトチーム DōGA　かまた ゆたか

フレームソースのきわめて高度な使用法として，人体モデルなどの多関節構造体の記述方法を解説します。本格的な人体モデルは後編として，まずは，構造体の関数化の表現を簡単なモデルで理解しましょう。

はじめに

高津「かまたさん，REND.Xですけど，コプロを直接アクセスするようにしたら30%ほど高速化されましたよ」

かま「偉い！　もっともっと高速化するんだ」

高津「頑張ればもう少し速くなりそうですね」

かま「よし，REND.X 改め REND.XVI と名づける！」

＊

XVIも仲間入りして，CGAを始める人も新たに増えるんじゃないでしょうか。CGAコンテストに出品するにしても，ちょっとしたストーリー作品を制作する場合でも，キャラクターとして人物やロボットなどが必要になってきます。しかし，このようにたくさんの関節によってつながっている物体（多関節構造体）は，FFEで扱うことができないため，エディタでフレームソースを書かなければいけません。でも，具体的にどのように記述すれば，簡単にモーションデザインできるのでしょうか？　以前にも多関節構造体の基本的な考え方については触れましたが，あの知識だけではとうてい作品を作ることはできません。

前々から，一度は本格的に取り上げなければいけないなとは思いつつも，あまりにハードな内容のため，ついついあと回しになっていたのが，今回の多関節構造体，つまりロボットバトルの作り方です。とにかく複雑で，フレームソースのありとあらゆる奥義が出てきます。できるだけ具体例（フレームソースリスト）を掲載しながら解説していきますので，頑張って理解してください。

量的にもかなりになりますので，来月号と2回に分けて掲載しますが，最終的には，ネットなどで入手した“歩く”とか“阿波踊り”というモーションデータを使って，オリジナルロボットや，Z，パロレイバーなどを自由に動かせるようにしたいと思います。

1．シッポを振るための復習

さぁ，どんどんいきましょう。

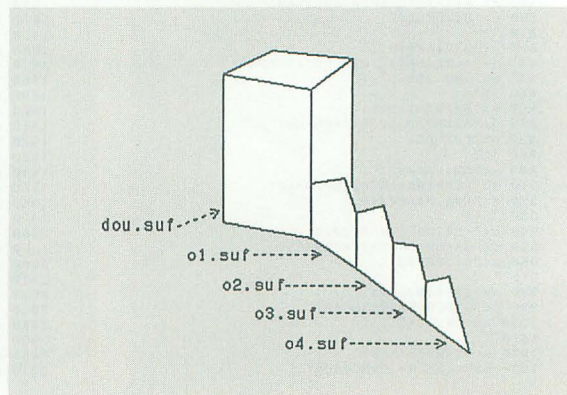
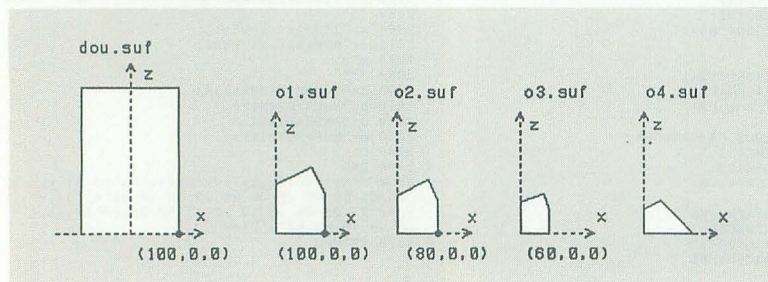
まず，簡単な多関節構造の例として，復習を兼ねて，ハコジラのシッポを動かしてみます。この例では，シッポは，左右にしか振らない（つまりZ軸にしか曲がらない）ものとします。

ハコジラのシッポは，図1のように，o1.sufからo4.sufの4つのパーツに分かれており，図2のように，dou.sufにつながっています。この構造をフレームソースで記述すると，リスト1のようになります。また，シッポを振るという動きを記述すると，リスト2のようになります。なお，各リストの左端に付いている番号は，行数を示すもので，実際には書かないでください。

多関節構造は，“{”と“}”の階層の深さで表現することは，以前解説しました。12，15，18，21行目のmovは，親のパーツの座標系で，そのパーツが親のどの部分に接続しているか（つまり関節の位置）を表しています。また，13，16，19，22行目のrotzは，それぞれの関節での接続角度です。リスト2では，この値を40度～－40度まで変化させることにより，シッポを左右に振っています。

図1　各パーツの形と大きさ

図2　接続関係（test1.fsc）

dou.suf
o1.suf
o2.suf
o3.suf
o4.suf

2. シッポを関数化する

こんなに簡単な構造のフレームソースでさえ，やたらに長くなってしまいました。このハコジラ，1カットだけの出演なら許せますが，あちこちのカットで，シッポをフリフリ出現するなら，そのたびにズラズラと書くのはどう考えても面倒です。

そこで，リスト2の12～27行目までを別ファイルにして，必要なときは，そのファイルを呼び出してくっつける（includeする）ようにしましょう。それから，よく考えると，シッポの曲がり方は，各関節とも同じ角度なので，ひとつの変数で表現できるはずです。これが，いわゆる関数化です。

リスト3とリスト4をご覧ください。リスト3が，リスト2の12～27行目をo3.fscという別ファイルにまとめたもので，リスト4では，それを13行目で呼び出しています。リスト3は，その物体（たとえばハコジラ）についてひとつだけ作成すればよく，このハコジラが何カットに出現しようと，各カットは，リスト4のように簡単に記述できます。

リスト4およびリスト3をFFに通すと，リスト5のようなフレームファイルが生成されます。このフレームファイルは，リスト2のフレームファイルとまったく同じものになります（当然作画しても同じ絵ができる）。

それでは，それぞれのリストについて簡単に解説いたしましょう。まず，リスト3(o3.fsc)つまり関数部分ですが，1行目が“o”という関数の宣言で，18行目で終了しています。関数oの内容は，2行目から17行目で記述されています。

関数oには，rzという仮引数がひとつ与えられています。rzというのは，この関数におけるローカル変数名なので，どのような名前でもかまいません。関数oが呼び出されるとき，ここの部分に具体的な数値が書き込まれるのです。2行目から17行目は，ただのフレームソースですが，3，6，9，12行目で変数rzが使用されています。ですから，関数oが呼び出されたときの具体的な数値はここに代入されます。

次にリスト4をご覧ください。1行目のinclude文は，o3.fscを取ってきてつなげる役目をしています。このinclude

[FFの実行]

リスト3(samp3.fsc)，およびリスト4(o3.fsc)をFFに通すときは，

FF samp3［リターン］

とします。o3.fscは，samp3.fscの中で指定されているので，いちいち指定する必要はありません。ただこの例では，パスを指定していませんので，同じディレクトリ内でなければいけません。

リスト1〔samp1.fsc〕

```
 1:#frame( fno, 1, 1 )
 2:@4.2@
 3:fram
 4:{
 5:      light pal( rgb ( 1 1 1 ) -3 2 -4 )
 6:        { mov (  800 -400  500 ) eye deg( 60 )
 7:        }
 8:        { mov (    0    0  100 ) target
 9:        }
10:
11:        { mov (    0    0    0 ) obj dou
12:              { mov (  100    0    0 )
13:           rotz (  0)
14:           obj o1
15:               { mov (  100    0    0 )
16:                     rotz (  0)
17:                     obj o2
18:                         { mov (   80    0    0 )
19:                       rotz (  0)
20:                       obj o3
21:                             { mov (   60    0    0 )
22:                               rotz (  0)
23:                               obj o4
24:                             }
25:                     }
26:               }
27:           }
28:     }
29:}
30:#endframe
```

リスト2〔samp2.fsc〕

```
 1:#frame( fno, 1, 20 )
 2:@4.2@
 3:fram
 4:{
 5:      light pal( rgb ( 1 1 1 ) -3 2 -4 )
 6:     { mov (  800 -400  500 ) eye deg( 60 )
 7:     }
 8:     { mov (    0    0  100 ) target
 9:     }
10:
11:      { mov (    0    0    0 ) obj dou
12:           { mov (  100    0    0 )
13:             rotz (¥div(40,-40,1,20,fno)¥)
14:                obj o1
15:                   { mov (  100    0    0 )
16:                 rotz (¥div(40,-40,1,20,fno)¥)
17:                      obj o2
18:                          { mov (   80    0    0 )
19:                      rotz (¥div(40,-40,1,20,fno)¥)
20:                      obj o3
21:                          { mov (   60    0    0 )
22:                            rotz (¥div(40,-40,1,20,fno)¥)
23:                              obj o4
24:                            }
25:                    }
26:              }
27:           }
28:     }
29:}
30:#endframe
```

リスト3〔o3.fsc〕

```
 1:#func o( rz )
 2:               { mov (  100    0    0 )
 3:                      rotz ( ¥rz¥ )
 4:                      obj o1
 5:                           { mov (  100    0    0 )
 6:                           rotz ( ¥rz¥ )
 7:                           obj o2
 8:                             { mov (   80    0    0 )
 9:                            rotz ( ¥rz¥ )
10:                                   obj o3
11:                                       { mov (   60    0    0 )
12:                                   rotz ( ¥rz¥ )
13:                                   obj o4
14:                                 }
15:                           }
16:                    }
17:             }
18:#endfunc()
```

リスト4〔samp3.fsc〕

```
 1:#include "o3.fsc"
 2:#frame( fno, 1, 20 )
 3:@4.2@
 4:fram
 5:{
 6:       light pal( rgb ( 1 1 1 ) -3 2 -4 )
 7:       { mov (  800 -400  500 ) eye deg( 60 )
 8:       }
 9:       { mov (    0    0  100 ) target
10:       }
11:
12:       { mov (    0    0    0 ) obj dou
13:       #do ¥ o( div( 40, -40, 1, 20, fno) ) ¥
14:       }
15:}
16:#endframe
```

は，複数のフレームソースで使用する部分を別ファイルにしておくときに便利です。たとえば，リスト4の6行目から10行目までの，光源，視点，注目点のデータをよく利用するのであれば，その部分をリスト6のような別ファイルにすることで，リスト4もリスト7のように短くなります。なお，関数は，必ず#frame以前に宣言しないといけませんので，関数をincludeするときは，通常ファイルの先頭で行います。

最後にリスト4の13行目ですが，#do文によって関数が実行されています。引数としてdiv(40, −40, 1, 20, fno) が与えられていますが，これはフレームナンバーが決まれば，あるひとつの値になりますから，ただの数値を与えていると考えるとわかりやすいでしょう。その数値がリスト3のrzに代入され，この13行目が，関数oの内容に置き変わるわけです。rzに代入される数値は，1フレーム目が40，20フレーム目が−40と，フレームごとに変わっていくことによって，リスト3の関節の角度も変わっていきます。

説明がわかりにくくて申し訳ありません。リスト3，4のフレームソースとリスト5のフレームファイルをじっくり見比べて理解してください。

3. シッポ関数の応用

多関節構造体の関数化の基本はとりあえず，ご理解いただけたでしょうか？　しかしまだまだ基礎。真の実用性を目指し，ちょっとずつレベルアップしていきましょう。

先ほど紹介したリスト3の関数oですが，￥〜￥の中にはrzだけでなくいろいろな式などが書けます。まずリスト8では，各関節での角度を変え，根元はあまり曲がらずに，シッポの先はよく曲がるようにしました。シッポではなく，何かの触角や朝顔のつるなどの場合，こちらの関数のほうがリアルな動きをします。

次に，ハコジラのシッポがおたまじゃくしのシッポのように，泳いで前へ進んでいるような動きの場合，リスト9のようになります。この場合，関数を呼び出すほうもリスト10のようにしておきます。

このシッポの動きは，sin関数を基本にしています。このとき注意しないといけないのは，FFは，rotのときは度が単位なのに，なぜかsin，cosなどはラジアン単位（360度＝2π）ということです。ですから，リスト10の9行目

リスト5[samp3.frm]

```
 1:/*********************    Frame    1    ***********************/
 2:fram
 3:{
 4:        light pal( rgb ( 1 1 1 ) -3 2 -4 )
 5:        { mov (  800 -400  500 ) eye deg( 60 )
 6:        }
 7:        { mov (    0    0  100 ) target
 8:        }
 9:
10:        { mov (    0    0    0 ) obj dou
11:              { mov (  100    0    0 )
12:                rotz (40.00 )
13:                obj o1
14:                    { mov (  100    0    0 )
15:                      rotz (40.00 )
16:                      obj o2
17:                          { mov (   80    0    0 )
18:                            rotz (40.00 )
19:                            obj o3
20:                                { mov (   60    0    0 )
21:                                  rotz (40.00 )
22:                                  obj o4
23:                                }
24:                          }
25:                    }
26:              }
27:        }
28:}
29:/*********************    Frame    2    ***********************/
30:以下省略
```

リスト8[o8.fsc]

```
 1:#func o( rz )
 2:          { mov (  100    0    0 )
 3:            rotz ( ￥rz/2￥ )
 4:            obj o1
 5:                { mov (  100    0    0 )
 6:                  rotz ( ￥rz/3*2￥ )
 7:                  obj o2
 8:                      { mov (   80    0    0 )
 9:                        rotz ( ￥rz￥ )
10:                        obj o3
11:                            { mov (   60    0    0 )
12:                              rotz ( ￥rz*3/2￥ )
13:                              obj o4
14:                            }
15:                      }
16:                }
17:          }
18:#endfunc()
```

リスト9[o9.fsc]

```
 1:#func o( rz )
 2:          { mov (  100    0    0 )
 3:            rotz (￥45*sin(rz)￥)
 4:            obj o1
 5:                { mov (  100    0    0 )
 6:                  rotz (￥45*sin(rz-3.14/3)￥)
 7:                  obj o2
 8:                      { mov (   80    0    0 )
 9:                        rotz (￥45*sin(rz-3.14/3*2)￥)
10:                        obj o3
11:                            { mov (   60    0    0 )
12:                              rotz (￥45*sin(rz-3.14/3*3)￥)
13:                              obj o4
14:                            }
15:                      }
16:                }
17:          }
18:#endfunc()
```

リスト6 [siten.fsc]

```
1:        light pal( rgb ( 1 1 1 ) -3 2 -4 )
2:        { mov (  800 -400  500 ) eye deg( 60 )
3:        }
4:        { mov (    0    0  100 ) target
5:        }
```

リスト7 [samp4.fsc]

```
 1:#include "o3.fsc"
 2:#frame( fno, 1, 20 )
 3:@4.2@
 4:fram
 5:{
 6:#include "siten.fsc"
 7:
 8:        { mov (    0    0    0 ) obj dou
 9:        #do ￥ o( div( 40, -40, 1, 20, fno) ) ￥
10:        }
11:}
12:#endframe
```

リスト10 [samp9.fsc]

```
 1:#include "o9.fsc"
 2:#frame( fno, 1, 30 )
 3:@4.2@
 4:fram
 5:{
 6:#include "siten.fsc"
 7:
 8:        { mov (    0    0    0 ) obj dou
 9:        #do ￥ o( fno/30*2*3.14 ) ￥
10:        }
11:}
12:#endframe
```

の式は，fnoが1～30で2π，つまり1周期となるという意味になります。

同様に，リスト9のrzには，ラジアン単位の角度が代入されます。sinの値は±1ですから，3，6，9，12行目では，それぞれ45を掛けることで，各関節は±45度だけ動きます。6，9，12行目でそれぞれ，π/3，2π/3，πを引いているのは，各関節の位相をずらすためで，30フレームで2πなのですから，π/3は5フレーム，つまり5フレームずつ遅れてシッポの動きが後ろの関節に伝達されていくわけです。

最後に，ちょっと無理のある設定ですが，“ハコジラが美しいワルツを聞いて，思わずシッポでリズムを取り出した”という動きを表現してみましょう。ポイントは，シッポが三角形上を動くので，ひとつの関数では書けず，場合分けをするという点です。

例によって，リスト12(sampb.fsc)から，リスト11(ob.fsc)の関数を呼んでいます。リスト12はほとんど一緒なのですが，9行目を見ると引数が2種類である点が異なります。一般に関数の引数の数はいくつでも問題ありません。

リスト11の1行目を見ると，2つの引数は，

frame ＝ fno　　angle ＝ 30

のように代入されています。frameはもちろんフレームナンバー，angleはシッポの最大角度を表しています。たとえば，シッポが動いていない状態から，だんだん大きく振れ始めるようなシーンの場合，この30の代わりに，

div (0 , 30 , 1 , 50 , fno)

を与えることで表現できます。

リスト11の2行目のabsは絶対値です。angleの値として負の数を与えられたときに，正の数に直してしまうわけです。

このように，都合の悪い数値を与えられても，関数側で訂正してやることで，関数をよりブラックボックス化できます。

たとえば，このハコジラのシッポが，デザインの都合上40度以上曲げられないときは，

```
#if( angle > 40 )
    #do ¥ angle = 40 ¥
#endif
```

という3行を加えます。

3行目の％は，余りを与える演算子です。timeには，frameを30で割った余りが入りますので，timeは0～29を周期的に繰り返すことになります。この使い方は非常に便利で，この関数を利用する元のフレームソースが何フレームあろうとも，その長さ分だけ，自動的に同じ動きを繰り返してくれます。

4行目が問題の場合分けで，timeが0～9の場合が5～21行で，10～29の場合が23～44行です（図3）。後半は，Z軸回転だけでなく，Y軸回転を入れることで，シッポを持ち上げています。24行目の変な関数は，シッポを直線的に動かさないためにcosを用いて複雑なことをしているだけで，あまり意味はありません。

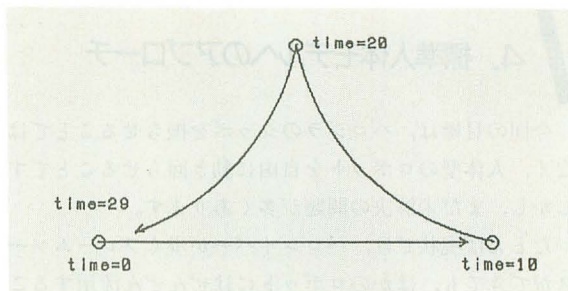



図3　シッポの動き

以上がシッポ関数の応用ですが，関数化というものをご理解いただけたでしょうか。関数のファイルもフレームソースですから，相当複雑なことが書けますし，関数から，さらにほかの関数を呼ぶこともできます。ようするに関数化は，ややこしいところをまとめて別ファイルにブラックボックス化して，複数のフレームソースから，いくつかのパラメータを与えるだけで簡単に使えるようにしようというものなのです。

リスト11〔ob.fsc〕

```
 1:#func o( frame , angle )
 2:#do ¥ angle = abs( angle ) ¥
 3:#do ¥ time  = frame % 30 ¥
 4:#if(time<10)
 5:#do ¥ rz = div(-angle , angle , 0 , 10 , time ) ¥
 6:            { mov (  100    0    0 )
 7:              rotz ( ¥ rz/2 ¥ )
 8:              obj o1
 9:                   { mov (  100    0    0 )
10:                     rotz ( ¥ rz/3 ¥ )
11:                     obj o2
12:                          { mov (   80    0    0 )
13:                            rotz ( ¥ rz ¥ )
14:                            obj o3
15:                                 { mov (   60    0    0 )
16:                                   rotz ( ¥ rz*3/2 ¥ )
17:                                   obj o4
18:                                 }
19:                          }
20:                   }
21:            }
22:#else
23:#do ¥ rz = div( angle , -angle , 10 , 30 , time ) ¥
24:#do ¥ ry = -angle *( 1-abs( cos( div(0,3.14, 10,30,time))))¥
25:            { mov (  100    0    0 )
26:              rotz ( ¥ rz/2 ¥ )
27:              roty ( ¥ ry/2 ¥ )
28:              obj o1
29:                   { mov (  100    0    0 )
30:                     rotz ( ¥ rz/3 ¥ )
31:                     roty ( ¥ ry/3 ¥ )
32:                     obj o2
33:                          { mov (   80    0    0 )
34:                            rotz ( ¥ rz ¥ )
35:                            roty ( ¥ ry ¥ )
36:                            obj o3
37:                                 { mov (   60    0    0 )
38:                                   rotz ( ¥ rz*3/2 ¥ )
39:                                   roty ( ¥ ry*3/2 ¥ )
40:                                   obj o4
41:                                 }
42:                          }
43:                   }
44:            }
45:#endif
46:#endfunc()
```

リスト12〔sampb.fsc〕

```
 1:#include "ob.fsc"
 2:#frame( fno, 1, 50 )
 3:@4.2@
 4:fram
 5:{
 6:#include "siten.fsc"
 7:
 8:        { mov (    0    0    0 ) obj dou
 9:        #do ¥ o( fno , 30 ) ¥
10:        }
11:}
12:#endframe
```

4. 標準人体モデルへのアプローチ

今回の目標は，ハコジラのシッポを振らせることではなく，人体型のロボットを自由に動き回らせることです。しかし，まだ未解決の問題が多くあります。

たとえば現状では，パロレイバーが歩くフレームソースができても，ほかのロボットにはぜんぜん流用することはできません。また，パロレイバーを別のカットで別の動きをさせようとした場合，同じような記述を何度も書き直す必要があります。さらに，フレームソースの記述の仕方，関数や物体の名前の付け方は，制作者ごとにまったく異なり，制作者以外の者にはさっぱり理解できません。

そこで，任意の多関節構造体は無理だとしても，最もよく使用される人体型の多関節構造体のフレームソースについて，一般的な表記方法を示し，ほかの人が作ったデータの有効利用ができるようにしたいと考えました。

- ポーズや動きのデザインがしやすいように記述する
- “歩く”とか，“ラジオ体操”といった動きのデータを，どのロボットでも利用することができるように記述する
- 動きのデータと各ロボットの形状デザインに依存する部分をはっきりと分け，各ロボットに依存する部分は別ファイルにし，ブラックボックス化する（そのロボットの制作者以外の人は，見る必要をなくしてしまう）
- 各シーンごとに記述しなければいけない部分を減らす
- パーツや関数の名前の付け方に規則を設け，制作者以外の者が見てもすぐわかるようにする

＊

まず，標準的な人体モデルのパーツと構造ですが，図4のようにしました。これだけのパーツに分かれていれば，人間の動きの大部分は記述できます。しかし，ハコジラのシッポと比べると，ものすごく複雑です。ハコジラのシッポの場合，その関数に必要な引数は，1つか2つぐらいでしたが，この標準人体モデルではなんと，引数が64個必要になります。……なんか気の遠くなるような数値ですね。これら引数をずらずら並べるのはちょっと現実的ではありません。またたとえば，16個目の引数が右腕のヒジの関節のy軸の値で，35個目の引数は……なんてことを覚えられるわけがありません。こういった問題を少しずつ解決していきましょう。

5. 形状デザイン上の注意点

残念ながら，そろそろページ数がなくなってきました。標準人体モデルを実際に動かすのは，来月にしましょう。そこで，来月までに皆さんご自分の形状データを制作し

関数の応用の応用

本文で掲載した多関節構造体の関数化はモーションデザインの例として解説しましたが，これを形状デザインに応用することができます。

何か複雑な形状でも，単純な形状の組み合わせで構成されていることはよくあります。残念ながらCADでは部品の概念がなく，複数の部品を伸ばしたり，回転させたりしてくっつけていくということができません。

そういった場合，“伸ばしたり，回転させたり”というのをフレームソースで記述して別ファイルに関数化し，メインのフレームソースからは，あたかもひとつのオブジェクトのように扱うことができます。

Graphic Galleryの桜の絵をご覧ください。この桜の花は，図のように花びら，ガク，雄しべ，雌しべからできています。リスト1(HANA.FSC)が，花を関数化したファイルです。2行目から6行目まで，花びらを72度ずつ回転させて置いています。9行目から14行目は雄しべですが，スケールや向きを変えて適当に散らしているわけです。

リスト2(HARU.FSC)は，その関数を利用した例で，Graphic Galleryの絵のフレームファイルですが，1行目でHANA.FSCをincludeし，11，14行目で呼び出しています。18行目が普通の物体（1枚の花びら）の場合ですから，ほとんど同じように扱えるというのがおわかりいただけると思います。

FFEで使用できないという欠点がありますが，FFEでは全体の形のシンプル版を用意して，あとでエディタで置換すれば問題ありません。CADで複雑な物体が作れないとお悩みの方は，ぜひ一度試してください。

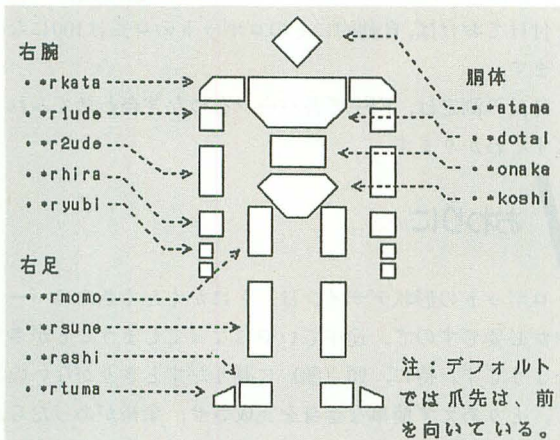
リスト1〔HANA.FSC〕

```
 1:#func hana()
 2:{ mov(  0 100   0)              rotx( 10) obj bira}
 3:{ mov(-80   0   0) rotz(  72) rotx( 11) obj bira}
 4:{ mov(-40 -50   0) rotz( 145) rotx(  9) obj bira}
 5:{ mov( 80  10   0) rotz( -72) rotx( 11) obj bira}
 6:{ mov( 40 -50   0) rotz(-144) rotx( 12) obj bira}
 7:{ mov(  0   0 -50) rotz(  20) scal(1.3 1.3 1.3 ) obj gaku  }
 8:{ mov(  0   0 -80)            scal(0.7 0.7 1.2 ) obj mesibe}
 9:{ mov(  0   0 -30) rotz(  90)                    obj osibe }
10:{ mov(  0   0 -20) rotz( -60)                    obj osibe }
11:{ mov(  0   0 -30) rotz(  10) scal(1   1   0.9 ) obj osibe }
12:{ mov(  0   0 -30) rotz(-150) scal(1   1   0.95) obj osibe }
13:{ mov(  0   0 -30) rotz( 160) scal(0.7 0.7 0.7 ) obj osibe }
14:{ mov(  0   0 -30) rotz(  40) scal(0.8 0.8 0.8 ) obj osibe }
15:#endfunc()
```

リスト2〔HARU.FSC〕

```
 1:#include "hana.fsc"
 2:#frame( fno, 1, 1 )
 3:@4.2@
 4:fram {  light pal( rgb ( 1 1 1 ) -3 -2 -4 )
 5:      { mov( 1000     0     0 )  eye deg( 60 )
 6:      }
 7:      { mov(    0     0     0 )  target
 8:      }
 9:
10:      { mov(    0  -400  -250) rotz( 50) roty(30) rotx(10)
11:      # do ¥ hana() ¥
12:      }
13:      { mov(-1500   800   500) rotz( 90) rotx(-50)
14:      # do ¥ hana() ¥
15:      }
16:         ・・・  中略  ・・・
17:      { mov(-7000   500 -2000) rotz(-60) roty(-60) rotx(-60)
18:        obj bira
19:      }
20:}
21:#endframe
```

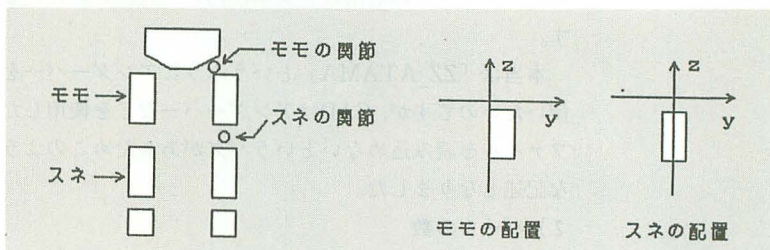
ておいてください。

CADはいまだに使いこなせないという方もいらっしゃるでしょうが，なんでしたら図4のようにほとんど直方体だけの形状でも結構です。しかし，以下の点には注意してください。

0）基本

各パーツごとに別の形状ファイルにしておくことと，各パーツの原点(0,0,0)の位置がそのパーツの関節の位置にしなければいけないことはすでにご存じのとおりです。モモのパーツとスネのパーツが同じ形状だとしても，関節の位置が違えば別の物体にしなければいけません（図5）。

1）各パーツの名前の付け方

図4　標準人体モデル

図5　関節位置とパーツの配置

ただの雑談

XVI使用感

新登場のXVIですが，もう入手した方も多いでしょう。期待が大きかっただけに，物足りないという感想もよく聞きますが，個人的にはけっこう気にいってます。なにしろ，16MHzになっても，ワープロ，ゲーム，ミュージック，パソ通などにはそれほど劇的な変化はありませんが，CGの場合もろに恩恵をうけるからです。特に大量の画像を必要とするCGAでは，CPUのパワーアップはありがたいかぎりです。でも，CPUが1.6倍になったからといって，周辺の都合でそのままの速度が出るとはかぎりません。実際，CGAにとってどれだけの速度が出せるか独自に調査してみました。

・レンダリング速度

旧FLOAT2.X使用時	1.61倍
新FLOAT2.X使用時	2.31倍

・アニメーション速度　1.61倍

おっ，見事に速くなってますね。うむ，広告に偽りなし。レンダリングテストは，コプロなしで，RENDで，VOYAGERを1枚作画させたものですが，FDへの書き込み時間なども含まれているので，計算速度自体はもう少し速くなっているといえるでしょう。新しいFLOAT2.Xも，PDSのFLOAT2A.Xと比べても1.2倍ほど高速化されているようです。

アニメーションは，もともと一定速度（20フレーム毎秒）にするためにタイムウェイトをかけているので，XVIでそのまま実行しても20フレーム毎秒のままですから，上の値はウェイトをはずした場合です。つまり，XVIでは，多少複雑な絵でも，毎秒約50フレームのアニメーションが可能です。ただし，もともとビデオが毎秒30フレームしかないので，まったく無駄ですけど……。

CGAコンテスト裏話

読者の皆さんは，CGAコンテストの作品集ビデオ(VHS，90分)を，もう申し込みました？　締め切りは5月末だから，まだの人は急いでください。申し込み方法は，郵便振替で，口座番号「大阪3-109598」，加入者名「DōGA」，金額は2,000円＋カンパ（任意）です。なお，ご自分の住所，電話番号，また通信欄に“ビデオ希望”と明記してください。

＊

さて，そのCGAコンテストだが，準備は想像以上に大変なのだ。その，ほんの一部でも紹介しよう。たとえば，“コンテスト当日の朝来て，上映用のVTRをセットする”，ただそれだけ。どう考えても簡単なことに思えるのだが，それを実際に行うと……。

スタッフがワゴン車3台で，YAMAHAホールに到着したのは，まだ朝9時であった。機材等を積み降ろし，ワゴン車は予定されていた駐車場へ向かった。各自が担当の仕事の準備に取り掛かったとき，上映班から最初のトラブルが報告された。VTRがない！

ワゴン車の座席の下に置いてあったのに気が付かず，駐車場へ行ってしまったのだ。VTRは，当日故障した場合のことを考えて2台持ってきていたが，両方とも積んだままという。しかし，20～30分もすれば，運転手が戻ってくるだろうから，取りに行かせればいい。

……だが，1時間たっても運転手は戻ってこなかった。道に迷う可能性を心配して，YAMAHAホールへ来る途中，一度駐車場に行って，位置を確認しておいた。さらに，先頭のワゴン車には地図を持ったナビゲータまで付けているのだ。それから1時間後，やっと運転手が戻ってきた。彼がいうには，駐車場からホールへは簡単に行けるが，逆は一方通行で行けない。さんざんウロウロしたあげく，やむを得ず駐車場の前の角を右折したところ，たまたまいた警官に右折禁止で捕まってしまったという。なんと運の悪い話だ。

しかし，なんとかVTRは手に入り，セッティングが完了した。そして，オープニングアニメーションから上映テストを行ってみた。妙に画質が悪い？　そして，突然止まってしまった。調べると，マスターテープが巻き込まれていた！

VTRを分解し，巻き込んだテープを丁寧に外してみたが，頭10秒分はグチャグチャで，どう見ても使用不能であった。オープニングアニメの問題は置いといて，とりあえず，もう1台のVTRをセッティングし，準備を進めた。しかし，2台目のVTRは，まったく映らない！　持ってきたVTRが2台とも故障したのだ！

メカに強い者が，VTRを分解し，修理を始める。同時に，今からVTRを入手する方法や，上映開始時間を遅らせること検討する。修理班の結論は，1台はまったく手が付けられないが，巻き込んだほうはなんとか動くようにはなる。しかし，いつ再び巻き込むかはまったくわからないという……。そのころになると，受賞者の皆さんがリハーサルのために集まっていた。もちろんリハーサルどころではない。もう“湾岸情勢のため上映会は自粛しました”と張り紙をして，大阪に逃げ帰ろうという案まで出された。ところが，運よく受賞者のひとりが秋葉原のある電気屋の社長と知り合いで，VTRを貸してくれるという。急いで車で取りに行くことになった。

確か，VTRが届いたころには，もうロビーにはお客さんがいたと思う。新しいVTRを接続し，オープニングアニメの途中からダビングしたところで，ぶっつけ本番，上映会はスタートした！

だが，トラブルはまだ続く。途中から始まるオープニングアニメを上映し，審査員の紹介が終わったところでそのVTRに問題があることがわかった。一時停止や再生するたびに，スクリーンに“一時停止”“再生”の文字が大きく映し出されるのだ。ちょっとこいつは恥ずかしい。

そこで，映写班は，私が主催者からの挨拶をいっている間（スクリーンは何も映さない）に，再び元のVTRに取り替えてしまった。そしてもうあとは，テープが巻き込んだらハイ終わりの状態で上映会を続けたのであった。

そのほか，マイクの数が足りないとか，東名高速道路でトラックとぶつかるとか，お金が足りなくなって，ビデオを何本以上売らないと帰れなくなったとか，トラブルは数知れない。このように，コンテストの裏ではたくさんの苦労があったことを知って，作品集のビデオを見れば，感激もひとしおだろう（？）。

図4をよく見ると，各パーツの名前に＊がついています。この＊には，皆さんが自由に決めたロボット（あるいは人体モデル）名を入れてください。つまり，たとえば「ZZ」という名前のロボットを制作した場合，頭のパーツは「ZZATAMA」となり，腰のパーツは「ZZKOSHI」となるわけです。

ただし，ファイルの識別は8文字以内という制限があります。あまり長い名前をつけると，パーツごとの識別ができなくなります。パーツは頭2文字で識別できますので，ロボットの名前は6文字以内にすればいいわけです。

本当は「ZZ_ATAMA」というふうにアンダーバーを使いたいのですが，CADはアンダーバーなどを使用したファイルを読み込めないというバグがあるためこのような記述となりました。

2）パーツの数

図4では，指がひとつのパーツだと省略しても，全部で22のパーツが必要になります。しかし，これは厳格なものではなく，これ以上でも，以下でも対応することができます。詳しくは来月解説しますが，自分のデザインしたロボットでは，足に指は付いていないので，爪先のパーツは必要ないとか，おなかのパーツと胴体のパーツは一体化しているというのはぜんぜん問題ありません。ただし，足が4本だとか，下半身が戦車のようにキャタピラになっているなどというのは，標準人体モデルとはいえません。

また，左右対称の物体の場合，片腕，片足を作るだけで結構です。

3）身長の測定

各パーツができましたら，ロボット全体の身長を測定してください。これも詳しくは来月ですが，基本的に標準人体モデルの身長は100ということになったからです。もちろん，身長がちょうど100になるようにデザインしなければいけないということではなく，たとえば身長が2480であった場合，関数のファイルの先頭に，

scal（0.04 0.04 0.04）（：0.04=100/2480：）

を付けておけば，自動的にそのロボットの身長は100になります。

身長の測定は，FFEで各パーツをつなぎ合わせてみればすぐわかります。

おわりに

ロボットの形状デザインは，とにかくたくさんのパーツが必要ですので，途中でいやになってしまうことが多いようです。特に，頭（顔）に凝りだすときりがないので，とりあえず簡単な全身を完成させ，余裕があったら，各パーツをバージョンアップさせていきましょう。来月には，このロボットが歩いたり戦ったりするんだと自分自身をはげまして頑張ってください。

さて，もう5月も終わりですが，皆さんの所属するクラブなどには，たくさん新入部員が集まりましたか？当チームの場合，大阪大学コンピュータクラブなどを通じて新スタッフが入ってくるのですが，例年，人数だけは相当数集まります。しかし……。

プロジェクトチームDōGAでは，ただいま女性マネージャーを募集しています。

プロジェクトルームが大阪市東淀川区にありますので，市内，阪急沿線，およびその近辺の方。高校生以上であれば年齢などは問いません。とにかく，この劣悪なる環境からスタッフを救うために力を貸してください！　お手紙お待ちしています。

柚姫の明るい悩み相談室

最近お便りが減って元気がありません。元気なお便りをお待ちしています。

そこで，今回は，3月2日，東京にて開催されたCGAコンテスト発表会の会場でご記入いただきましたアンケートから，ほんの一部紹介させていただきます。

このコンテスト，姫ももちろんスタッフとして参加しています。でもほんとにたいへんだったんですよ。高速道路ではトラックに体当たりされるわ，宿の門限に間に合わなくなるわ，当日になってビデオが壊れてしまうわ……。

姫自身は何をしていたかというと，実は，なんと司会という大役を果たしていたのです。それについては後ほどに……。

＊

アンケート：当コンテストに対する要望，感想

>古いビデオも再配布してほしい。

姫：残念でした。もうなくなってしまいました。来年，こんなことにならないように今回のコンテストのビデオは早めに入手しておいてくださいね。実費(2,000円)＋カンパです。と，しっかりSHOW・BY，ショーバイ!!

>面白い！　ぜひ年2，3回やってください。

姫：こらこら，こっちの身にもなってよ。年2，3回もトラックにつっこまれてたまるもんですか。

>コンテストとして，幅広い門を設けてください。

姫：確かに狭かったですねぇ。でも，YAMAHAホールを勝手に改装するわけにはいかないんですけど。

>いろんな人の取り組みが見られて，次回も楽しみです。

姫：来場所も来てね。

>「おたく」雰囲気をなるべく除きましょう。できたら女の子もいっぱい来るようなコンテストがいいなぁ。

姫：まったくだ。来年のコンテストにはぜひ女の子をたくさん連れてきてください。スタッフが涙を流して喜びます？

アンケート：CGAシステムをお持ちの方は，バージョンをご記入ください

>3.00

姫：実費＋カンパを送りますので，大至急DōGAまで送ってください。スタッフが涙を流して喜びます（ちなみに最新バージョンは2.23です。3.00はまだ影も形もないのだよ）。

アンケート：どのような作品がお好みですか？

>モダン焼き

姫：あんた関西人やな。

>燃えるものならなんでも！

姫：せっかくの作品を燃やさないで！

アンケート：司会について

>ドツキ漫才がよかった。もっと漫才やってください。

>女性の司会者の台本ボウ読み，ダイコンがとっても気に入った。

>次回の上映会でもぜひあの2人に司会をしていただきたいです。

>いくら関西系の団体だからといって，司会が漫才だとは……。

>司会がヘタ。

姫：ぷるぷる，えーい，あのときの司会の女性は，私だと知ってのロウゼキか!?　あれはすべて台本だったんだ〜い。普段の姫は男の人をたたいたりしないよぉ〜，うるうる。姫はおとなしいいい子です。文句のある奴はかかってきなさい！

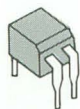
（来年は司会なんかやらないぞと心に誓う姫）

ハードウェア工作入門《12》

メカトロニクス制御 その2

Misawa Kazuhiko
三沢 和彦

今回はステッピングモーターの解説だけでなく，ロジック回路を設計するための考え方もあわせて説明していきます。また，製作に自信のない人は，前回の解説編を参照しながら理解を深めて，自分で回路製作を行うときに役立てましょう。

前回はステッピングモーターの原理とその駆動法について解説しました。ステッピングモーターはコンピュータ制御に適していることは前回説明しましたが，CMOS IC 1個の簡単な回路で駆動できてしまいます。論理回路自体は簡単なのですが，汎用のTTL ICを使った回路は第1回の基本I/O回路以来ですので，目的のロジック回路を設計するときの考え方もあわせて解説してみようと思います。また，今回の回路ではトランジスタやダイオードといった個別部品も使っていますので，それらの扱い方も学んでいきましょう。

ステッピングモーター駆動回路の設計

ステッピングモーターの動作は5月号をもう一度復習してください。その動作原理から考えて，モーターを回転させるには，モーターのA,A′,B,B′の4相に順番に電流をONしていくようなスイッチを組んでやればよいということがわかります。そのON/OFFをジョイスティックポートのOUTで直接コントロールするというのもひとつの手段なのですが，不幸なことにジョイスティックポートのOUTは3ビットしかありません。とはいえ，実際のステッピングモーターの駆動にはA,A′,B,B′の4相は区別せず，1ステップずつ1クロックを送って回転させてやるだけで済みます。そこで，コンピュータから送り出すクロック信号を独立した4ビットに振り分けてやるようなインタフェイスを組む必要があるわけです。このように，一般的なインタフェイス設計のコンセプトとして覚えておいてほしい手順は，

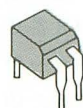
1) 外部機器をコントロールするために入出力される必要のある信号をひととおり列挙する。ここでは，外部機器とはステッピングモーターで，入出力するべき信号というのは4相のON/OFFである。

2) それとはまったく別に，外部機器をコントロールするためにコンピュータが入出力するのに，最低限必要な信号は何かを考える。ここでは，ステップごとのクロック信号1ビットとモーターの回転方向を決める制御信号1ビットの計2ビットである。このとき，4相別々に4ビットと考えるのは間違いである。

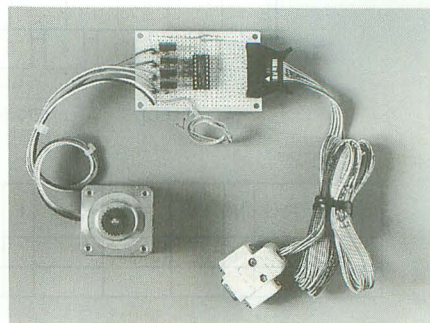
3) インタフェイスとしては，外部機器側に必要な信号とコンピュータ側に必要な信号とのギャップを埋め合わせるものとして設計していく。

以上の手順です。

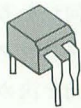
このようにして設計されるステッピングモーター駆動回路は，1ビットクロックを入力するごとにONの位置を1相ずつシフトさせていけばよいわけで，それには1クロックを入れていくたびに4ビットのパラレル出力がシフトしていくような「シフトレジスタ」という回路を使ったものが一般的です。このシフトレジスタでは，シフト方向を制御する端子もあり，これによって回転方向を制御することができます。

しかし，今回の回路ではもう少し別の視点から設計してみました。ステッピングモーターの4相に順番に0,1,2,3と番号を付け，コンピュータからの出力をクロックの代わりに2ビット整数で0→1→2→3をそのまま出してやるのです。回転を逆にするときは3→2→1→0の順に出力すればよいのです。この方法でも制御信号は2ビットでOKです。あとは，2ビット整数を独立の4ビット出力に振り分ける回路を設計すればよいことになります。この目的には，デコーダというロジックICがぴったりです。

デコーダICの選び方

以上に述べた方針でデコーダを使うとして，今回はHC238を選んでみました。そこでまず，今回の回路の中心となるHC238というICについて動作を規格表で確認することから始めましょう。図1はTTL IC規格表の抜粋です。このHC238は3→8デコーダと呼ばれるものです。入力ABCが3ビットの2進数（0～7）となっているのに対して，出力はY0からY7までの8ビット出力になっています。そして，入力した値に対応したビット番号のビットだけHレベルにして，残りの7ビットをすべてLレベルにするのです。

実際の回路では2ビットしか使わないので，最上位ビットのC（3番ピン）はLに落としておいて，入力はA入力とB入力（1,2番ピン）だけで処理します。すると，たとえば入力に順番に0から3までを与えていくとそれにしたがって，出力のビット0からビット3までが順番にONになっていくのです。そこで，モーターの4相をビット0からビット3にまでひとつずつ対応させておけば，最初の目的どおりA相→B相→A′相→B′相→A相の順に励磁してモーターを回転させることができるのです。

出力は15番ピンから7番ピンまで（8番ピンはGND）の8ビットですが，モーターの駆動に使うのは15〜12番ピンで，ほかの使わないピンには何もつながないでおきます。一般に，何も使わない出力端子はそのまま開放しておくのが基本です。その代わり，使用しない入力端子を放置しておいては誤動作を起こしてしまうので，必ずHかLかに設定しておきます。

このほか，4〜6番ピンはイネーブル端子といって，このHC238全体のON/OFF制御端子です。「イネーブル」という言葉は直訳すると「可能にする」という意味です。この端子によって，ICを使用可能にするかどうか制御するのです。規格表を見てもわかるとおり，イネーブルをOFFにすると入力にかかわらず，出力はすべてLになってしまいます。

さて，このデコーダICはHCシリーズのCMOSタイプを使っていて，いつものLSシリーズのTTLタイプとは違う品種です。これは，規格表でデコーダを探すとほとんどが負論理で，今回の回路には適しません。すなわち，よく使われているLS139やLS138では，入力した値に対応したビット番号のビットだけLレベルにし，残りの7ビットをすべてHレベルにするのです。これでは，あとで述べるトランジスタスイッチのON/OFFには都合が悪いのです。そこで，正論理のデコーダを選ぶとすると最も適当なのはHC238ということになるのです。なお，LS238というのは製造されていませんが，規格にしても値段的にもHCタイプはLSタイプとほぼ同等なので，問題ありません。

個別部品の選び方

では次に，トランジスタ周辺の部品の選び方を考えてみましょう。そのためには，ステッピングモーターそのものの定格を再度チェックする必要があります。図2を見ると駆動電圧が6.3V，コイル抵抗が9.7Ωとなっていますので，単純にオームの法則で駆動電流値を見積もると，650mAとなります。それに対し，HC238の出力は4mAしか流せないので，直接駆動することはできません。そこで，トランジスタで電流を増幅する必要があるのです。このトランジスタはもちろんモーターの駆動電流の650mAを流せるものを選ばなければなりません。さて，トランジスタスイッチの回路は電力用NPNトランジスタのエミッタをGNDに落とし，コレクタにインタフェイス回路とは独立に駆動電源電圧をかけておき，ベース・エミッタ間に電流を流すとコレクタ・エミッタ間に大きな電流が流れる（スイッチON）ようにしています。電力用トランジスタとしてトランジスタ規格表を探していくと，2SD686，687が見つかりました。この規格を図3に示します。この回路における電流増幅率（hFE）も最大で2000近くあり，ベース電流がHC238からの4mAのときは，理論的には0.004×2000＝8Aまで余裕があることになります。実際には最大コレクタ電流が4Aなので，それ以上流すとトランジスタが壊れてしまいますが，そ

図2　ステッピングモーター

〔SIBAWRA ENGINEERING WORKS CO,LTD〕

型番：S4H40B06F-01
ステップ数：200（1.8DEG/STEP）
駆動電圧：6.3V
コイル抵抗：9.7Ω
静止トルク：400g（推定）
出力トルク：1kg（推定）
最大応答周波数：1000PPS（推定）

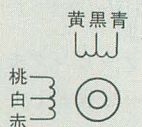
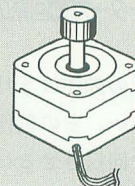
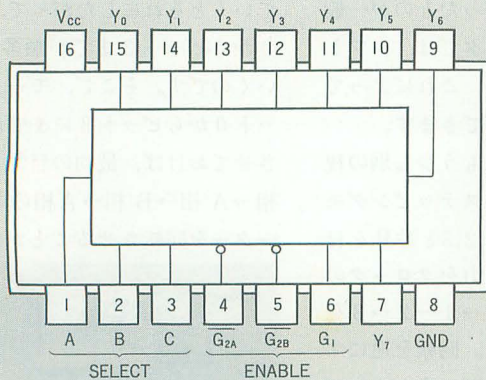
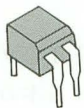



図1　HC238

出力電流特性		N	LS	ALS	ALS 1000	F	S	AS	AC	HC	HCT	単位
全出力	H→									4		mA
	L←									4		mA

入力					出力							
ENABLE		SELECT										
G_1	$\overline{G_2}$*	C	B	A	Y_0	Y_1	Y_2	Y_3	Y_4	Y_5	Y_6	Y_7
X	H	X	X	X	L	L	L	L	L	L	L	L
L	X	X	X	X	L	L	L	L	L	L	L	L
H	L	L	L	L	H	L	L	L	L	L	L	L
H	L	L	L	H	L	H	L	L	L	L	L	L
H	L	L	H	L	L	L	H	L	L	L	L	L
H	L	L	H	H	L	L	L	H	L	L	L	L
H	L	H	L	L	L	L	L	L	H	L	L	L
H	L	H	L	H	L	L	L	L	L	H	L	L
H	L	H	H	L	L	L	L	L	L	L	H	L
H	L	H	H	H	L	L	L	L	L	L	L	H

$\overline{G_2 *} = \overline{G_{2A} + G_{2B}}$
H：ハイレベル　L：ローレベル　X：H or L

れでもコイル駆動のための650mAにも十分余裕があります。このように，電力用トランジスタを選ぶときには最大コレクタ電流と電流増幅率とがポイントになります。

ダイオードについても，最大許容電流が問題となりますが，ここでは一般的によく用いられている１Ａ整流用の10D1を選びました。これにはほとんど選択の余地はありません。また，定電圧ダイオードは逆電圧16ＶのRD16Fにしました。定電圧ダイオードの役割についても５月号を参照してください。これによって，コイルの誘電起電力が16Ｖを超えると，トランジスタに負担がかからないように誘導電流を逃がしてやることになります。このとき，トランジスタに常にかかる電圧は，モーターの電源電圧の６Ｖに誘導起電力の16Ｖを加えた22Ｖになっています。しかし，2SD687の最大コレクタ電圧は80Ｖですから，トランジスタは壊れることはありません。以上のように，個別部品を選択するのも特別難しいわけでは決してなく，ほんの少しの知識で使いこなせるようになるのです。

製作実習

部品表を表１に示します。コネクタおよび基板はいつも使っているものと同じです。半導体部品はいつもどおりT-ZONEパーツショップで入手しました。ステッピングモーターは秋月電子で格安品を購入しました。特に型番指定はしませんでしたので，皆さんが使うのも４相ステッピングモーターならばどんなものでもOKでしょう。ただし，駆動電圧とコイル抵抗のチェックは忘れないようにしてください。駆動電流が500mA程度のものがあればベストです。

では，配線に移りますので，実体配線図（図４）を参照してください。配線はこれまでに説明してきたとおり，まず主な部品を基板上に配置します。基板用コネクタ，ICソケット，抵抗，トランジスタ，ダイオードの順に取り付けていくのがよいでしょう。ICソケットは3，4，5，8番ピンはGNDに直結ですので，ピンを内側に折り曲げてGNDラインとともにハンダ付けしてしまいます。同じように16番ピンを内側に折り曲げて＋5Vラインにハンダ付けします。

次に抵抗４本を配線します。スペースの都合で抵抗は立てて取り付けることにしました(図５)。抵抗の取り付けには，片方は

表１　部品表

品名	数量	価格
IC用基板（サンハヤトICB-87）	1枚	90円
10ピン基板用コネクタ（HIF3BA10P-DS）	1個	100円
ステッピングモーター（芝浦S4H40B06F-01）	1個	400円
16ピンICソケット	1個	30円
HC238	1個	90円
2SD687	4個	@100円
10D1	4本	@20円
RD16F	4本	@90円
1kΩ抵抗	4本	
ACアダプタ（6V500mA）	1個	500円
ACアダプタジャック	1個	100円
ビニール配線材	少々	

図３-a　2SD687

電極接続		
1	2	3
ベース	コレクタ(フランジ)	エミッタ

図４

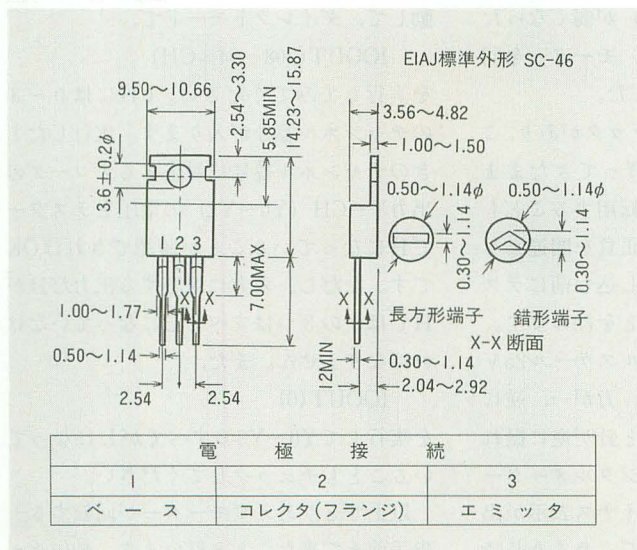
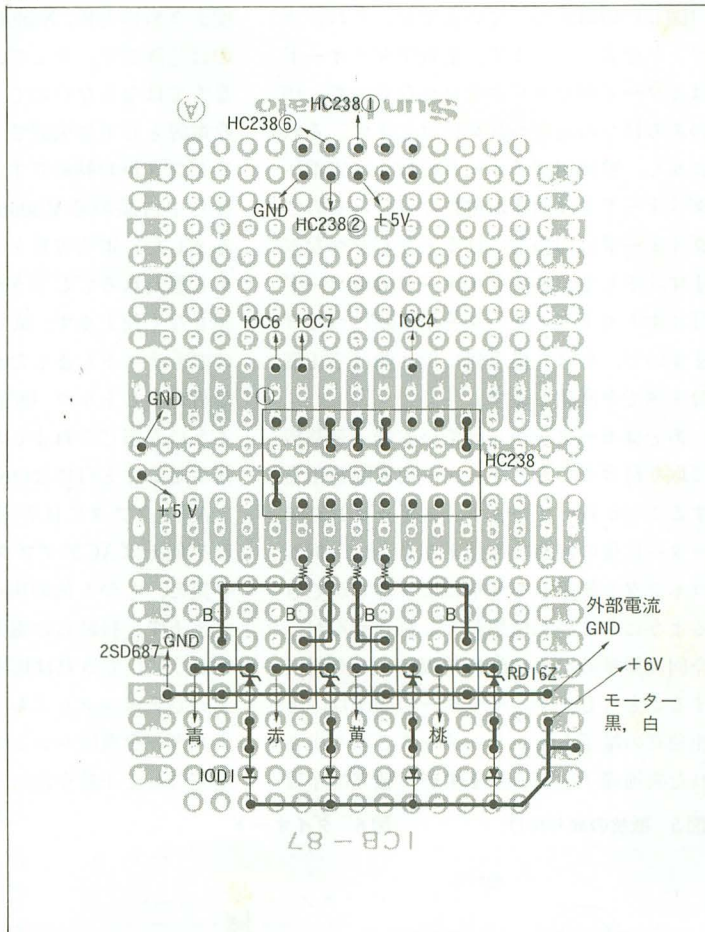
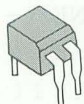



図３-b　2SD687

型名	社名	用途	構造	最大規格 (T_a=25℃) V_{CBO} (V)	V_{EBO} (V)	I_C (mA)	P_C (mW)	T_j (℃)	電気的特性 (T_a=25℃) I_{CBO}最大値 (μA)	V_{CB}(V)	直流またはパルスh_{FE}	V_{CE}(V)	I_C(mA)	バイアス V_{CB}(V)	I_E(mA)	h_{fe} h_{fb}*	h_{ie} h_{ib}* (Ω)	h_{re} h_{rb}* (×10⁻⁴)	h_{oe} h_{ob}* (μ℧)	f_{ab} f_T* (Mc)	C_{ob} (pF)	$r_{bb'}$ h_{ie}(real)* (Ω)	外形	備考
2SD687	東芝	PA. SW	Si. E	60	5	3A	25W (T_c=25℃)	150	20	60	>2000	2	1A			t_{on}=0.1μS, tf=0.2μS t_{stk}=1μS							268	ダーリントン

ICの端子に直結ですが，反対側は抵抗の足を折り曲げ，そのままトランジスタのベースまで伸ばしてハンダ付けします。同じように，トランジスタの足も折り曲げて配線の助けとします。トランジスタの足は正面から見て左からベース，コレクタ，エミッタになっていますので，図を見て間違えないようにしてください。ベースの足は先ほどの抵抗の足にハンダ付けします。エミッタの足は，横方向に1列に折り曲げて4個分すべて直結します。これはGNDラインになりますので，ICのGNDラインともつなぐ必要がありますが，それには別のビニール被覆線でジャンプさせてつなぎます。これをジャンパ線といいます。

コレクタにはダイオードとステッピングモーターとをつなぎます。ダイオードの極性も間違えないように確認してください（図6）の印がついていますが，それがカソードを表しています。定電圧ダイオードはカソード側をコレクタにつなぐので，印のあるほうの足をハンダ付けします。それに対し，整流用ダイオードのほうは定電圧ダイオードとは向きが逆になっているので，ダイオードは印のないほうどうしをつなぎます。そして，整流ダイオードのカソードは4本ともすべてモーター用電源につなぎますので，やはり横1列に折り曲げて4個分すべてを直結します。

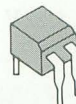
あとはモーターから出ている端子を基板に取り付けるのと，モーター用電源を用意することが残っています。ステッピングモーターは他の回路にも使えるように，何かコネクタを使用して簡単に取り外しができるようにするのが理想かもしれませんが，今回は簡単のために直接基板にハンダ付けすることにします。モーターの端子は，4相独立の端子とセンタータップで取り出された共通端子とがありますが，独立端子は順番にトランジスタのコレクタにつなぎます。4相のA相→B相→A′相→B′相の順にHC238のビット0,1,2,3の順に対応するようにつなぐことを間違えないでください。今回使ったステッピングモーターでは端子が色分けされていますので，その順番を守ってください。

最後に，ビニール線でコネクタとICソケットの端子を電源ラインも含めて5本のジャンパ線で配線すれば，基板のほうは完成です。

モーター駆動用電源

モーター用の駆動電源はステッピングモーターの定格にあわせて準備しなければなりませんが，今回使ったモーターには6V 1A程度の電源が必要です。電流がある程度大きめのため，X68000内部から取り出すのは危険です。そこで，外部電源を用意しなくてはならないのですが，いちばん手頃な電源としては家庭でもなじみのあるACアダプタがお勧めです。今回は秋月で見つけた中古品の6V500mAのものを使ってみました。電流容量としては少し足りなめかと思われるでしょうが，実際には何の問題もなく使えます。試しに6V300mAのものでもテストしましたが，電流が小さめのためか少しトルク（回転力）が弱くなったようにも感じられましたが，モーターを回転させること自体はOKでした。

ACアダプタには専用ジャックがあり，これを使えばACアダプタは買ってきたままで使え，しかも他の用途に転用することもできます。接続には電源の正負を間違えないように，できればICを差し込む前にテスターでチェックしておくことを勧めます。テスターを電圧レンジでフルスケール25V程度にして正負を決めます。万が一，逆につなぐと針が逆に振れる（デジタルメーターではマイナス表示が出る）ので，あまり長時間逆につなぐことはやめてください。さて，ジャックの接触点には，内側の軸と外側の端子とがありますが，どちらがプラスでどちらがマイナスかというのはACアダプタによって違っていることが多く，ジャックだけでは決められないので，その都度表示で確認しなければなりません。

図5 抵抗の取り付け

図6 ダイオード

動作チェック

動作チェックをする前に配線チェックをもう一度しましょう。間違いやすいのは，

・トランジスタの足を間違える
・ダイオードを逆に付ける
・モーターの端子の順番が違う
・ACアダプタの正負が逆

です。トランジスタスイッチ周辺の素子がごちゃごちゃしていて注意が必要でしょう。

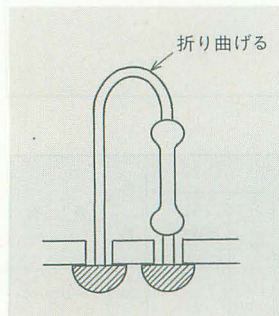
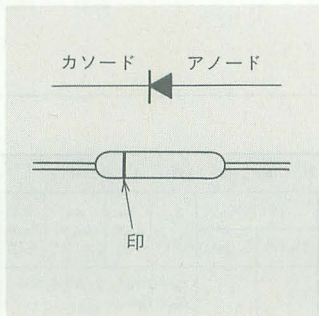
デコーダ回路のチェックは，BASICを起動して，ダイレクトモードで，

IOOUT(208−64×CH)

を実行してみてください。CHには0〜3のチャンネル番号が入ります。実行したときのチャンネル番号に対応するデコーダの出力Y−CH（Y0〜Y3）の電圧をテスターでHになっていることが確認できればOKです。ただし，入力に対応する出力だけがHでほかの3つはすべてLになっていなければなりません。また，

IOOUT(0)

を実行してY0〜Y3のすべてがLになっていることもチェックしてください。

以上でとりあえずモーターが回転する一歩手前まで来たことと思います。動作チェックのためのサンプルプログラムは次回に載せるつもりです。さらにまた，応用プログラムとして，コンピュータでメカを制御する実例を予定していますので，ぜひ次回までに駆動インタフェイスを完成させておいてほしいと思います。

BACK ISSUES バックナンバー案内

ここには1990年6月号から1991年5月号までをご紹介しました。現在1990年11，12，1991年1～5月号までの在庫がございます。バックナンバーおよび定期購読の申し込み方法については，188ページを参照してください。

1990

6月号（品切れ）
特集　創刊8周年記念PRO-68K（付録5"2HD）
Oh!Xアンケート結果大分析大会
連載　ショートプロぱーてぃ/Z80's Bar/PurePASCAL
X-BASIC調理実習/X68000マシン語プログラミング
●X1turbo用コマンドシェルシミュレータ
●ハードウェア工作入門
LIVE in '90　ナイトアームズ/悪魔城伝説/この木なんの木
THE SOFTOUCH　三国志II/FAR SIDE MOON/グラナダ
全機種共通システム　X68000用S-OS"SWORD"他

7月号（品切れ）
特集　マシン語への第一歩
X68000SUPER-HD試用レポート
連載　ショートプロぱーてぃ/Z80's Bar/DōGA・CGA
X-BASIC調理実習/PurePASCAL
●INTEGRAL X1――ノーマルX1への対応
●ハードウェア工作入門
LIVE in '90　夢幻戦士ヴァリスII/トッカータとフーガニ短調
THE SOFTOUCH　サーク/あーくしゅ/ダウンタウン熱血物語
全機種共通システム　リロケータブルアセンブラWZD

8月号（品切れ）
特集　ADVANCED 2D GRAPHICS
100号記念特別モニタプレゼント
連載　ショートプロぱーてぃ/Z80's Bar/INTEGRAL X1
X-BASIC調理実習/X68000マシン語プログラミング
PurePASCAL/ハードウェア工作入門
●X68000用画像回転プログラム　XROT0.X
LIVE in '90　OMENS OF LOVE/ENDLESS RAIN/ダートフォックス
THE SOFTOUCH　大航海時代/ウルティマV/プロミストランド
全機種共通システム　リンカWLK

9月号（品切れ）
特集1　日本語を処理するための序章
特集2　ADVANCED 2D GRAPHICS
連載　ショートプロぱーてぃ/Z80's Bar/DōGA・CGA
X-BASIC調理実習/マシン語プログラミング
PurePASCAL/ハードウェア工作入門
●清水和人流プログラミング道場
LIVE in '90　風の谷のナウシカ/ラジオ体操第一
THE SOFTOUCH　T&T/D-Again/シムシティー/ギャラガ'88ほか
全機種共通システム　BILLIARDS

10月号（品切れ）
特集　電子音楽術入門
連載　ショートプロぱーてぃ/Z80's Bar/DōGA・CGA
マシン語プログラミング/ハードウェア工作入門
清水和人流プログラミング道場
●荻窪圭の大人のためのX68000
●中森章のようこそここへC言語
LIVE in '90　Rise And Fall/PARADOX/キューピー3分クッキング
THE SOFTOUCH　ワールドコート ルーンワース/闇の血族/提督の決断
全機種共通システム　ライブラリアンWLB

11月号
特集　理科系のGAME REVIEW
連載　Z80's Bar/DōGA・CGA/カードゲーム
マシン語プログラミング/ハードウェア工作入門
PurePASCAL/X-BASIC調理実習
ようこそここへC言語/INTEGRAL X1
荻窪圭の大人のためのX68000
LIVE in '90　ピラミッドソーサリアン/ザ・スキーム
THE SOFTOUCH SPECIAL　ラグーン/幻獣鬼/サイバリオン/GUNSHIP他
全機種共通システム　スクリーンエディタEDC-T

12月号
特集　XCのための傾向と対策
連載　X-BASICプログラミング調理実習/ハードウェア工作入門
マシン語プログラミング/ショートプロぱーてぃ/Z80's Bar
大人のためのX68000/ようこそここへC言語/INTEGRAL X1
●シミュレーションプログラミング入門
●特別企画アナログジョイスティックの製作
LIVE in '90　グラディウスIII/メタルサイト
THE SOFTOUCH SPECIAL　イメージファイト/ジェミニウイング/NAIOUS他
全機種共通システム　STACKコンパイラ

1991

1月号
特集　急接近！SX-WINDOW
特別付録　謹賀新年PRO-68K（5"2HD）
連載　ハードウェア工作入門/シミュレーションプログラミング入門
DōGA・CGA/ショートプロぱーてぃ/大人のためのX68000
PurePASCAL/清水和人流プログラミング道場/X-BASIC調理実習
LIVE in '91　めぞん一刻/涙で綴るパパへの手紙
THE SOFTOUCH　ソル・フィース/銀英伝II/続ダンジョン・マスター他
製品紹介　光磁気ディスクCZ-6MO1
全機種共通システム　ライブラリアンWLB

2月号
特集1　グラフィックの"実験的"手法
特集2　SX-WINDOWプログラミング
連載　ハードウェア工作入門/シミュレーションプログラミング入門
マシン語プログラミング/大人のためのX68000/Z80's Bar
ショートプロぱーてぃ/INTEGRAL X1/ようこそここへC言語
●1990年度 GAME OF THE YEAR ノミネート発表
LIVE in '91　Misty Blue/スプーンおばさん
THE SOFTOUCH　栄冠は君に/KLAX/ダイナマイト・デューク他
全機種共通システム　ダイスゲームKISMET

3月号
特集　MIDI & MUSIC PROCESSING
連載　ハードウェア工作入門/シミュレーションプログラミング入門
マシン語プログラミング/大人のためのX68000/Z80's Bar
ショートプロぱーてぃ/DōGA・CGA/C言語/PurePASCAL
●SXLIFE完結編/ウィンドウシステム大比較
●周辺機器新製品紹介
LIVE in '91　戦いの兜/LITTLE WING/リゾ・ラバ/花
THE SOFTOUCH　アトミック・ロボキッド/スペースローグ他
全機種共通システム　アクションゲームMUD BALLIN'

4月号
特集　人とゲームのインタフェイス
連載　DōGA・CGA/シミュレーションプログラミング入門
ハードウェア工作入門/ようこそここへC言語/Z80's Bar
ショートプロぱーてぃ/清水和人流プログラミング道場
●新連載 吾輩はX68000である/よいこのSX-WINDOW講座
●決定！1990年度GAME OF THE YEAR
LIVE in '91　Easy Come, Easy Go!/シシリエンヌ
THE SOFTOUCH　メルヘンメイズ/中華大仙/スライス他
全機種共通システム　SLANG用カードゲームDOBON

5月号
特集　新登場！ X68000XVI/XVI-HD
特別付録　黄金週間PRO-68K（5"2HD）
第6回　言わせてくれなくちゃだワ
連載　ハードウェア工作入門/ようこそここへC言語
大人のためのX68000/X68000マシン語プログラミング
ショートプロぱーてぃ/マシン語カクテルin Z80's Bar
LIVE in '91　ブーピーキッズ/NO. NEW YORK
THE SOFTOUCH　マーブル・マッドネス/シグナトリー/石道他
全機種共通システム　実数型コンパイラ言語REAL

みんなで狙い撃ち!

Komura Satoshi 古村 聡

illustration : T.Takahashi

今月のショートプロはゲーム2本立て。X1用反射型アクションゲーム「LASER」とX68000用ダーツゲーム「DARTS」です。そして，今月からぱーてぃハンズ第3部開始の予定だったんですが……。予告はあります。

皆様いかがおすごしでしょうか？　(で)です。今月採用の投稿原稿を読んでいて思い出したんですが，この本が出るころにはもう新学期が始まって1カ月になるんですね。4月に学校や会社に入られた皆さんもそろそろ生活に慣れ始めたころでしょうか。なかには，X68000を下宿に持ち込んで念願の優雅なX68000のある独り暮らし！　なんてのを始められた方もいらっしゃることでしょうね。

ええ，ええ。独り暮らしにX68000ってのはいいもんですよね。誰にも邪魔されずにいくらでもゲームだって，プログラム作りだって，原稿書きだって，誰にも邪魔されずにできるはずですもんね。そう，ちゃんと友達さえ選べば……。

頼むからうちを溜り場にして，X68000でよってたかってゲームをしないでくれえ。ちったあ，俺に原稿かかせてくれよお……，しくしくしくしくっ……。友達選びを間違えて溜り場のオーナーになってしまわないように気をつけましょうね，これから独り暮らしを始める皆さんは。ぐっすし。

LASER

DARTS

恐怖のレーザー光線

ではでは，仕事のほうにかかりましょう(しかたがないからノートパソコンで……)。今月の1本目は東京都(あ，大学に受かって引っ越したんですね。おめでとう)の坂本康さんで，X1用アクションゲーム「LASER」です。

LASER for X1

(CZ-8FB01)

東京都　坂本　康

ゲームが始まると画面上部中央からレーザーが伸びていきます。壁に当たると跳ね返るので，ジョイスティックのトリガを使って壁をうまく作り，エネルギーが切れる前にレーザーの先端（自機）を攻撃目標に導いてください。

ジョイスティックのトリガボタンを押すとレーザーの先端（＋）に壁ができます。つまり，普通にトリガを押すと＋の軌跡上に壁を作っていきます。壁と壁のすきまに＋が来たときには通り抜けていってしまいます(図1)。トリガと一緒に上下左右にジョイスティックのレバーを倒すと，＋のとなりにも壁を作ることができ，すきまのない壁となります。

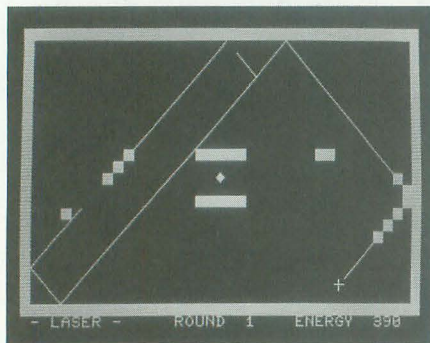
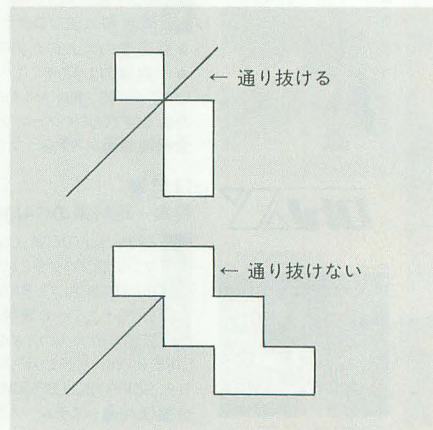
でも，壁を作れるからといってあんまり作りすぎちゃうと，今度は逆に壁に埋もれて身動きとれなくなっちゃいます。とりあえず，リターンキーを押すことでブロックは壊せるんだけど，そのたびにエネルギーを300も使っちゃうからね。使いすぎに気をつけましょう。

え，なになに？　“トリガを1個しか使わないでリターンキーを使ったので，操作性が悪いとかいわれそうですが，これには深い理由があります。先日，ジョイパットをばらしたら，ボタンが1個なくなってしまったんです。僕の開発コンセプトは，

自分の都合に合わせる

というものですので自分にないトリガ2は使わなかったのです。すべてのゲームが白黒画面なのも単に自分の都合に合わせてそうなった，というだけです”

なるほど，私もリターンキーじゃなくてトリガ2に爆破を割り当てたほうがいいんじゃないかな，とは思ったんですけどね。投稿するときには本当はこういうことはやらないほうがいいんだけど，ね。というわけで，不便に思った人は自分の都合のいいように作りかえましょう。プログラムを見てどこを変えればいいか考えなきゃいかんわけだからプログラムの勉強にもなりますよ。早速，リターンキーからトリガ2に

図1

変えるとか，キーボードでプレイできるようにしてみてはいかがでしょうか。簡単ですからね。

さてさて，坂本さんはこれでショートプロに載ったプログラムが4本目で“夜中に……”の太田敬三さんにならんでショートプロ掲載タイ記録になるのですね（と，投稿原稿に書いてあったんだ，これが)。常連の皆さんも，これから常連の皆さんもがんがん投稿して記録を塗りかえていっていただきたいのでした。

みんなで燃えろ

では，続いて今月の2本目。X-BASICのゲーム「DARTS」です。

DARTS for X68000

(X-BASIC)

京都府　安岡毅

1人から4人までで楽しめる，ダーツの301というゲームです。多人数でやるとやたらと燃えるゲームなんだ，これが。

301というのは,ダーツのなかでもわりと知名度の低いゲームなのですが，プレイヤーが順番にダーツを投げてポイントした得点を,はじめの持ち点である301点から引いていき，最初にちょうど0点になった人の勝ちというものです。

プログラムをRUNするとはじめにプレイヤーの名前を聞いてきます。1人目から順番に名前を入力してください。4人いないときはリターンキーだけを押すと，そのプレイヤーは飛ばされ，いないことになります。名前を入力しおえるとゲームスタートです（ゲームの展開上，順番が早い人のほうが多少有利になるので，そのへんは多少配慮して順番を決めましょう)。

マウスを左右に動かすと画面左上の矢印が動きます。これで狙う方向を決めてマウスをカチッとクリック。次はパワー。左下のパワーゲージが動きますのでパワーを決めます。最後に高さ。画面右側のダーツが動きます。狙う高さを決めるとダーツが飛んでいって，プスッ！　やったね，ど真ん中，という具合です。ちなみにボードの得点は真ん中から20,15,7,3,1点になっています。

このゲーム，“ちょうど0点になると勝ち”っていう条件が曲者なんですよね。たとえば，最後に3点とか余っちゃうと，うまく3点にぴったり当てないと点が減らないから，ゲームが終わらなかったりするのだな。

おかげさまでせっかくいちばんに点数がひと桁になったのに，3点が取れなくてもたもたしていて，20点残ってたやつにズバッと真ん中に当てられて逆転されてしまったりして……。くく，くやしい。で，ついついもう1ゲーム，なんて……。

リスト1　LASER

```
1 'LASER  Ver.1.2    Programmed by  Y.Sakamoto     April 1,1991
10 WIDTH 40:INIT:DEFINT A-Z:SOUND6,25:SOUND11,0:SOUND12,30
20 R=1:S=0:E=1500
30 CLS:LINE(0,0)-(39,23),"■",B:LOCATE1,24
40 PRINTUSING"- LASER -     ROUND###     ENERGY#####",R,E;
50 FORI=0TOR:LOCATE INT(RND*39),INT(RND*24):PRINT"■■";:NEXT
60 LOCATE17,10:PRINT"■■■■■":LOCATE19,12:PRINT"◆"
70 LOCATE17,14:PRINT"■■■■■":U=1:V=1:X=20:Y=1:C$="＼"
80 LOCATEX,Y:PRINTC$
90 IF INKEY$(0)=CHR$(13) THEN E=E-300 ELSE 120
100 SOUND7,55:SOUND8,16:SOUND13,0:LINE(X-1,Y-1)-(X+1,Y+1)," ",BF
110 LINE(0,0)-(39,23),"■",B:LOCATE19,12:PRINT"◆":PAUSE5
120 E=E-1:LOCATE33,24:PRINTUSING"#####";E;
130 IF SCRN$(X+U,Y+V,1)<>"■" THEN X=X+U:Y=Y+V:GOTO190
140 AU$=SCRN$(X+U,Y,1):AV$=SCRN$(X,Y+V,1)
150 IF AU$="■" AND AV$="■" THEN U=-U:V=-V:GOTO190
160 IF AU$<>"■" AND AV$<>"■" THEN U=-U:V=-V:GOTO190
170 IF AU$="■" THEN U=-U:Y=Y+V
180 IF AV$="■" THEN V=-V:X=X+U
190 A$=SCRN$(X,Y,1):IF A$="◆" GOSUB"CLEAR":GOTO30
200 IF U*V>0 THEN C$="＼" ELSE C$="／"
210 IF A$<>" " THEN C$=" "
220 LOCATEX,Y:PRINT"+":IF STRIG(1) THEN C$="■" ELSE 250
230 S=STICK(1):M=(S=4)-(S=6):N=(S=8)-(S=2)
240 IF SCRN$(X+M,Y+N,1)<>"◆" THEN LOCATEX+M,Y+N:PRINT"■";
250 IF E>0 THEN 80 ELSE BEEP1
260 LOCATE0,24:FORI=0TO11:PRINT:NEXT:PRINTTAB(15);"GAME OVER"
270 FORI=0TO11:PRINT:NEXT:BEEP0:REPEAT:UNTIL STRIG(1):GOTO20
280 LABEL"CLEAR":C$="▒":SOUND7,55:SOUND8,15:FORJ=0TO1
290 FORI=0TO20:LINE(19-I,12-I)-(19+I,12+I),C$,B:NEXT
300 C$=" ":NEXT:SOUND8,16:SOUND13,0
310 CONSOLE:PAUSE10:LOCATE12,11:PRINTUSING"ROUND###  CLEAR",R
320 LOCATE26,24:PRINTUSING"ENERGY #####",E;
330 PAUSE10:EN=1500-R*100:R=R+1:IF EN<500 THEN EN=500
340 WHILE EN>9:EN=EN-10:E=E+10:LOCATE33,24:PRINTUSING"#####",E;
350 WEND:PAUSE15:RETURN
```

リスト2　DARTS.BAS

```
10 /*DARTS GAME
20 screen 1,1,1,1:console ,,0
30 dim int sco(3):dim int ply(3)={48,64,80,96}:dim str NAM(3)
40 int x,y,bl,br,co,dx,dy,po,k,han,sc:str na[8]
50 SETTEI():MAIN():end
60 func MAIN()
70 k=0:repeat:na="":sco(k)=301:input"player name";na
80 if na="" then NAM(k)="NOTHING" else NAM(k)=na
90 k=k+1:until k=4
100 cls:locate 32,1:print"DARTS GAME"
110 for i=0 to 3:locate 32,i+3:print NAM(i);sco(i):next
120 locate 3,29:print"POWER|":locate 36,29:print"|"
130 locate 63,14:print"-":mouse(0):mouse(2):mouse(4):k=0
140 while 1:dx=495:dy=495:po=70:co=120
150 repeat:if NAM(k)="NOTHING" then k=k+1
160 if k>3 then k=0
170 until NAM(k)<>"NOTHING"
180 sp_move(1,co,240,1):sp_move(2,240,ply(k),2)
190 sp_move(3,dx,dy,2):sp_move(4,po,470,2)
200 repeat:msstat(x,y,bl,br):if x>0 and co<256 then co=co+2
210 if x<0 and co>0 then co=co-2
220 sp_move(1,co,240,1):until bl<>0:for i=0 to 1000:next
```

ええい！　だから，私に仕事をさせろ，原稿を書かせろといってるじゃないか！人がわざわざほかのマシンで仕事してんのに多人数ゲームだからってひきずりこむなよお。わし，明日締め切りなんだからさあ……，しくしくしくっ！

ということで，また来月。

教訓。なによりまず友達を選ぶべし。

(で)のぱーてぃハンズ第3部
(予告)

ええ，まったくもうしわけありません。今月号から第3部“思考ゲームを作るぞ！”が始まるはず……，だったのですが，実はプログラムを作るのが間に合わなくて，いきなり予告編という情けない事態になってしまったのです。シオシオ。

で，作るゲームなのですがだいたいは決まっています(当たり前)。ゲームの名は思考ゲーム，「選択二十五（仮）」。選択二十五はその名のとおり，16から－9までの数の入った25個のマス目の中から，あるルールにしたがってお互いに数字をひとつずつ取っていき，その数字を足した合計が大きいほうが勝ち，という単純なゲームなのですが，なかなかに遊べるものにはなるはずなのであります。

で，プレイヤー1は人間がやって，プレイヤー2はX68000にやらせて，えっちらおっちらと考えさせます。ちなみに，思考ルーチンはすべての思考ルーチンの基礎（といわれているかもしれない），ミニマックス法を使える……といいなあ，などと考えているので対戦型思考ゲームやシミュレーションの基礎になる……かもしれない(かなり弱気だなあ)。もしかして，予定は未定の可能性もないではないが，とりあえず，待て次号！

```
230 co=co+8:repeat:msstat(x,y,bl,br)
240 if po>270 then po=70
250 po=po+3:sp_move(4,po,470,2)
260 until bl<>0:for i=0 to 1000:next
270 po=po-70:repeat:dy=dy-3:msstat(x,y,bl,br)
280 if dy<240 then dy=511
290 sp_move(3,dx,dy,2):until bl<>0 or dy<240
300 repeat:dx=dx-2:po=po-1:if po<0 then dy=dy+1
310 sp_move(3,dx,dy,2)
320 until dx<8 or dy>511:m_play()
330 apage(1):sp_move(5,co,dy-256,2):han=point(co,dy-248)
340 switch han
350 case 1:sc=20:break
360 case 2:sc=15:break
370 case 3:sc=10:break
380 case 4:sc=7:break
390 case 5:sc=3:break
400 case 6:sc=1:break
410 default:sc=0:break
420 endswitch
430 locate 0,14:print sc;" ":for i=0 to 3000:next
440 if sco(k)-sc>=0 then sco(k)=sco(k)-sc
450 locate 32,k+3:print"        "
460 locate 32,k+3:print NAM(k);sco(k)
470 if sco(k)=0 then break
480 k=k+1:if k>3 then k=0
490 endwhile
500 locate 10,15:print"PLAYER";k+1;"WIN!"
510 repeat:msstat(x,y,bl,br):until bl<>0:endfunc
520 func SETTEI()
530 dim char SP0(255)={0,0,0,0,0,0,0,1,1,0,0,0,0,0,0,0,
540                   0,0,0,0,0.0,1,1,1,1,0,0,0,0,0,0,
550                   0,0,0,0,0,1,1,1,1,1,1,0,0,0,0,0,
560                   0,0,0,0,1,1,1,1,1,1,1,1,0,0,0,0,
570                   0,0,0,1,1,1,1,1,1,1,1,1,1,0,0,0,
580                   0,0,1,1,1,1,1,1,1,1,1,1,1,1,0,0,
590                   0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,0,
600                   1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,
610                   0,0,0,0,0,1,1,1,1,1,1,0,0,0,0,0,
620                   0,0,0,0,0,1,1,1,1,1,1,0,0,0,0,0,
630                   0,0,0,0,0,1,1,1,1,1,1,0,0,0,0,0,
640                   0,0,0,0,0,1,1,1,1,1,1,0,0,0,0,0,
650                   0,0,0,0,0,1,1,1,1,1,1,0,0,0,0,0,
660                   0,0,0,0,0,1,1,1,1,1,1,0,0,0,0,0,
670                   0,0,0,0,0,1,1,1,1,1,1,0,0,0,0,0,
680                   0,0,0,0,0,1,1,1,1,1,1,0,0,0,0,0}
690 dim char SP1(255)={0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
700                   0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
710                   0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
720                   0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
730                   0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
740                   0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
750                   0,0,0,0,0,0,0,0,0,0,1,1,1,1,1,1,
760                   1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,
770                   1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,
780                   0,0,0,0,0,0,0,0,0,0,1,1,1,1,1,1,
790                   0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
800                   0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
810                   0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
820                   0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
830                   0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
840                   0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0}
850 sp_init():sp_clr():sp_disp(1):sp_on(1,5)
860 sp_def(1,SP0,1):sp_def(2,SP1,1):sp_color(1,rgb(31,0,0))
870 apage(0):vpage(15):circle(128,128,11,9,0,360,265)
880 for i=1 to 5:circle(128,128,i*23+11,9,0,360,265):next
890 fill(0,256,15,511,8):fill(0,279,15,486,7)
900 fill(0,302,15,463,8):fill(0,325,15,440,7)
910 fill(0,348,15,417,8):fill(0,371,15,394,7)
920 apage(1):fill(2,2,254,254,6):fill(25,25,231,231,5)
930 fill(48,48,208,208,4):fill(71,71,185,185,3)
940 fill(94,94,162,162,2):fill(117,117,139,139,1)
950 palet(0,rgb(0,10,0))
960 for i=1 to 6:palet(i,rgb(0,10,0)):next:palet(9,65535)
970 palet(7,rgb(31,0,0)):palet(8,rgb(0,31,0))
980 m_init():m_alloc(1,100):m_assign(1,1):m_trk(1,"@45v15c")
990 endfunc
```

X-OVER・NIGHT
（クロスオーバーナイト）

［第12話］ハイテクも使いよう

TAKAHARA HIDEKI 高原 秀己

情報化時代もすっかり板についたといわれている。複写機やファクシミリが企業に常備され，ワープロやパソコンが企業にも，さらに個人にも，どんどんと売れていくのを見ると疑う余地はない。

僕などは早くからパソコンを使っていたこともあって，結構情報化している人なのだろう。実際，取材のときに最新製品を手にする機会が多かったこともあって，プロフェッショナルな水準は別にすると，OA機器と呼ばれる機械はおおむね操作することができる。

だが，最近はちょっと考え方を基本に戻す作業をしている。既存メディアの特性を改めて考え直し，うまく使い分けなければ，という感覚になっている。

たとえば，作図，作表。

僕の会社で使っているワープロは作図とか，グラフ作成機能とかがついていて，「使ってやろう！」と意気込む人は多いのだが，うまく使えるのはレアケース。だいたいは無駄に時間を使ってしまうのがオチなのである。

次のような相談を持ちかけられるときがよくある。

「この表をワープロで作りたいんだけど，どうするんだっけ？」

そこで以下のように話を続けていく。

「数字だけ並べてみてください」

「数字だけじゃなくて，表を作りたいから聞いているんだってば」

「いいから，数字だけ文字間隔に余裕持たせて打ってください。できたら，いったん印刷してみてくださいな」

こういうとたいていはしぶしぶやる。印刷が終わったら，

「はい。それじゃ，定規を使ってきれいに線を引いて表にしてください。それをコピーすればいいんです。そのほうが，断然手早くできますから」

相手はポカンとして不満そうなのであるが，ワープロの作表や罫線機能を使おうが，定規で線を引こうが，最終的にはコピーするんだから同じこと。それならば，労力の少ないほうが得なのである。

ワープロ文書の特殊編集はハサミとコピー機ですむ場合が非常に多い。イメージスキャナを使って写真を文書に割り付けることができるようになっているのもあるが，なければ困るというわけではない。文書を作成してから，写真をコピーすればすむんだから。

これはワープロにかぎったことではない。たとえば，編集機能つきデジタルコピー機などというのがあって，操作が複雑でややこしい画像合成機能を使えば，2枚の文書を貼り合わせることもできる。

しかし，そんなものはハサミで貼り合わせる部分を切り抜いて，くっつけてからコピーすればいいことなのである。

だいたい，編集機能などはよほどのときを除けば使う必要はない。どうもOA機器にいろんな機能がついていると，せっかく買ったんだからその機能を使わなくてはならない，という気分になってしまう傾向が強すぎるようだ。

このように，本来は手作業や既存メディアで用がすむのに，わざわざハイテクツールに頼ってしまうケースは，情報収集においても見受けられる。

僕がいつも使っているパソコン通信は，いろいろなデータベースにゲイトウェイアクセスしてくれるサービスが備わっているので，過去の出来事だの，人物情報だのと，ついつい頼ってしまうクセがついている。

だが，パーフェクトなペーパーメディアが1冊手元にあれば，高い料金を支払って電子情報に頼る必要はそもそもないはずなのである。

決定的にこういう感覚にとらわれたのは，最近，ある新聞社年鑑を購入したからだ。個人でこの種の図書を購入している人はごく少ない。事実，僕の周りの人で買っている人などいない。

まあ，これは当然だろう。新聞社年鑑は会社の資料室なり図書館に最新の版が必ず常備してあることを僕たちはちゃんと知っている。僕の場合など，どこに置いてあって，どこに座って読むかということまで頭に入っている。それをわざわざ購入する必要は薄いのである。

百科事典の場合，家を新築したり転居したときに，記念というか，新しい置物として購入する人が結構多い。事実，ズラリと百科事典が書斎や応接間に陣取っていると迫力があるし，いかにも知的な家庭であるような印象も受ける。辞書にもそういう役割がある。

だが，同様の役割で新聞社年鑑を購入する人はまずいない。置物にしては薄いし，家財道具といったデザインが施されている図書でもないからだ。

いままでは，僕も月3,4回，わざわざ会社の資料室に行って，必要なページのコピーを取ってきて使う，という生活をずっと続けてきた。先日は単に気紛れで衝動的に買ってしまっただけなのである。

せっかく買ってきたのだからと丹念に最初から眺めていったのだが驚いた。新聞社年鑑とは，いわゆる「なんでも載っている本」だったのである。政治，経済から芸能，スポーツまで。政府から地方自治体，海外各国まで。政府調査も民間調査も合わせて，実にありとあらゆる統計やデータリスト，名簿が網羅されている。

この情報量でわずか5,000円足らずなのだから，恐るべきコストパフォーマンスともいえる。

ほかにも新聞の縮刷版，辞書，事典など，類似の図書は多い。最近，データブックの類がウケているようで，膨大な情報を収めた本がいろいろ発売されている。

いままで述べてきたことは，ちょうど，「歩けば5分で着く場所に，自動車で行ったら渋滞で15分かかってしまった」というような話と似ている。

CADと製図道具，作画ソフトと絵の具，ワープロとノート。いまの世の中，いろいろなメディアが溢れてはいるけれど，それに振り回されるのではなく，上手に使い分けることがますます必要になってきたということであろう。

第49回　知能機械概論――お茶目な計算機たち――

肥大したアザラシの群の中で

Macintosh専門誌

Macintoshのユーザーやファンを対象とする雑誌もだいぶ増えてきました。そうすると自然に，それらの雑誌を見る我々の目も，1,2誌しかなかったころとは違ってきびしくなってきます。その結果，マイナーチェンジを繰り返して，実際，息も絶えだえに思える雑誌もあります。

そうした状況で，「マックパワー」（アスキー）と「MacJapan」（技術評論社）の2誌は，比較的順調に発行を続けているようです。両誌とも同じ大きさ，値段であり，ちょっと見には似たような内容，あるいは編集方針ではないかと思います。

ところが少し読んでみると，それがそうでなく，まったく異なる性格を持っているということに気づかれることでしょう。まあ，出している会社の名前を比べてみて，そりゃそうだろうと推測した読者の方も多いかもしれません。

両誌の違いを際立たせる，ひとつの面白い例があります。昨年末に発売されたMacintoshの新製品，「Macintosh Classic/LC/IIsi」の取り上げ方がそうです。

「マックパワー」のほうは昨年の12月号で「ニューMacintosh 3機種の全貌」と題して，冒頭の39ページを費やして大々的に特集を組んでいます。それに対し，「MacJapan」のほうはMJExpressという枠で「アップル社，工場で新製品発表」（なんて渋いタイトルなんだ！）と題して，数ページにわたって事実だけを淡々とレポートしているだけです。

「MacJapan」はその後も，個性の強い連載記事群の中で，失望感のようなものを伴いながら紹介していて，特にプッシュして紹介したという記事はまったくといっていいほどありません。ひと言でいえば，黙殺です。これは，どう考えても新機種に対する不満をそのような形で表していると考えざるをえません。

一方，「マックパワー」のほうは，「全貌」を紹介したはずなのに，各機種を別々の号で再びレビューしています。それぞれ，「名実ともに時代の定番となるか，Macintoshの新生機」，「ホビーのみならずビジネスユースにも購入を勧められるマシン」，「高速化しつつコンパクトで低価格になるのは大歓迎」と，一見すると絶賛の嵐といった風潮です。

ただし，「マックパワー」の中でも座談会の記事では比較的本音が出ているようです。たとえば，「新機種をと考えているユーザーへの'91年の購入のアドバイス」（1991年2月号）などを読むと，Macintosh LCはエントリーマシンとしてはまあまあいいが，それ以外はほとんどパッとしないのでは，といった雰囲気が場を占めているのが感じられます。

異彩を放つ連載記事

「マックパワー」誌で（めずらしくも）Macintoshの新製品を具体的な形で批判している記述がありました。インダストリアルデザイナーである，川崎和男氏の連載「Design Talk」の中でのくだりです。

「Macintoshの造形に見られるカジュアル性をベースにしたフォーマルな品格は正当に認めるとしても，Macintoshの新シリーズの正面の局面的造形処理と側面に装飾的に施されたスリットの過剰さは，もっと考えられるべきであったのではないかと判断している」（文献1）。

川崎氏は折り畳み式車椅子や計算機制御ベッドの開発で第36回毎日デザイン賞を受賞しているデザイナーで，Macintoshの利用をデザイン界で啓蒙していることでも有名です。

連載は昨年9月号から始まっています。とにかく，川崎氏の意見は読む側にダイレクトに伝わってきます。そのことを切実に物語っている彼の文章をぜひ多くの人に読んでもらいたいと思います。

内容はMacintoshにとらわれたものではなく，大きな見地からのマシンのインタフェイスや，デザインの将来への方向付けを行っています。具体的なテーマとしては，フェティシズム（物神崇拝）の対象としてのMacintosh，マウスドローイングの持つやわらかなコミュニケーション，マルチメディアの衝撃などがあります。

これまでに，特に著者の思いが爆発しているように受け止められる記事が2回ほどありました。

ひとつめは，湾岸戦争が勃発したときに書かれた第6回です。その中で，デザイナーとしてデザインと戦争を考えたときに思想として確立すべきは何か，それは何がグッドデザインかという論理とその実証であると主張しています。戦闘機やミサイルの形をかっこいいと感じる感性を育むプログラム作りをデザインが果たすべきであるとまでいっているのです。デザインというものをここまで考えている人なのです。

そして具体的に，Macintoshこそそのようなアプローチの対象，あるいは道具とすべきであるとしています。対象としての位置付けというのは，Macintoshをモデルにした情報化論，道具論，文化論を築き上げながら，論議と具体的表現を伝道していくことであり，道具としての位置付けというのは，Macintoshを使うことによって恐ろしいものを具体的に表現する力を養うことを指しているのです。

デザイナーの怒り

そしてもうひとつ，川崎氏の連載第9回（ぜひ読んでみてください）こそ，彼のデザイナーとしてのたぐいまれな本質を見せつけた記事です。

ここで彼は現状のインダストリアル（工業）デザインへの辛辣な攻撃を行うとともに，デザイナーへ課された重い仕事を提示しています。

記事の中で，まず彼は「アフォーダンス」ということばを示します。それは，独創的な知覚論を展開した心理学者J.J.ギブソンの定義したことばであり，環境が提示したものという意味を持ちます。たとえば，雨が降ってきたときに，大きな木が「ここに来て雨宿りしませんか」といかにも呼びかけているように受け取れるような心理状況を指しているのです。

具体的な例として，オリベッティの2つのタイプライター，レッテラブラックとバレンタインに基づいて，モノにはフォーマル性とカジュアル性というアフォーダンスがあるのだと彼は述べます。

次に具体的に攻撃を開始します。PC-9801やFM-Rにはフォーマル性もカジュアル性もなく，結論的にいうならば美学というものがまったく欠落していると批判します。そして，そのデザイナーに対しては，

「あなたがたの犯した罪を償うべきだ」，ユーザーに対しては，「すでに美学なき大衆であることを自白しているようなものではないか」と詰め寄ります。

企業のデザイナーに，「肥大したアザラシ」のようなラジカセ（実際，それに対してＳ社が最近発売したきわめて律儀な直方体のラジカセは，たいへん魅力的です）や「バカげた形」のカメラの批判をしたそうです。デザイナーのひとりは，現代は大衆があのようなカーブを好んでいる，そして売れている，と笑って答えたそうです。そのとき，暴力が許されるのなら，そのデザイナーを殴ってやりたかったとまで彼は述べています。

川崎氏は記事の最後のほうで，デザインの果たすべき理想を，哲学と美学のない企業家に奪われてはならないとしています。彼の叫びが痛切に伝わってきます。他人や他人の仕事を単に過激に批判するのはそうむずかしいことではないかもしれません。しかし，彼の場合は違います。彼の叫びは彼自身へも重い義務として，ズシンとぶち当たっているからです。

邪悪なアフォーダンス

PC-9801が持つアフォーダンスについて語ろうとしても，そこには何もないようなマシンなのだということについては，本誌の読者にはあまり異論はないかもしれません。

しかし，アフォーダンスがないということ，つまり，モノが我々に何も語りかけてこないということは，かなり致命的であると思いますが，たちの悪い，極端にいえば邪悪なアフォーダンスを持つよりはまだましといえるかもしれません。

たちが悪いというのは，単にラジカセが「肥大したアザラシ」のようで品がないというような，表面的な問題をいっているのではありません。モノの持つアフォーダンスとそのデザインされるモノの所有者が誰か，という関係において示される性質のことをいっているのです。

工業製品からは少し離れますが，いい例があります。それは有名になった，また本誌読者にはお馴染みのデザインともいえる，あの新都庁舎です。

新都庁舎の持つイメージは近未来的で先端的なものを表すシンボルのようでもありますが，同時に下界に住んでいる庶民を威圧するような形態を持っているともいえます。

もし，あれがパソコンのフォルムのデザインというのならば，それなりのアフォーダンスを持っていると評価されるべきでしょう。しかし，それが都庁舎となると話は大きく違ってきます。なぜならば，ひと言でいえば，中央集権，あるいは中央集権的な権力誇示というものはもはや時代遅れであり，また時代遅れとなるべきものであるからです。

都庁舎のアフォーダンスという観点からではなく，「豪華な都庁舎」というレッテルづけによる（シャワーや大理石がどうしたという内容の）批判は，一時ずいぶん盛り上がったものです。しかし，都知事選では別の要素が支配するつまらない展開となり，結果的にはその問題はうやむやとなってしまったようで，その点に関しては誠に残念といえましょう。

雑誌と評価記事

デザイナーは売れている事実や売らんがための戦略をそのデザイン意図に取り込むべきでない，という川崎氏の主張はまさしく正論であり，同時に本質的に解決のむずかしい問題に関連しているでしょう。

新しく登場したマシン（あるいはそのデザイン）を評価する側（雑誌）がなるべく純粋な気持ちを持ち，説得力のある記事を提示するということは，そのような正論が通りやすくする土壌を作る点で影響は小さくないと思われます。いいものはいいとし，悪いものは具体的なアフォーダンスの記述とともに指摘する（それがむずかしいのであれば黙殺という態度もしかたないでしょう）。

ところが，実は雑誌の側にも，デザインにおける問題とまったく同様の構造が内在しているという点を見逃してはなりません。このことは先に触れた「マックパワー」誌の座談会の冒頭で，編集の人がいみじくもいっていることから容易に察しがつくことでしょう。

「12月号で新機種の特集を掲載したところ，読者からの反応はたいしたもので，アッという間に雑誌が売れたそうです」。

このような問題は本誌のような雑誌すべてに存在する普遍的な問題ともいえましょう。悪いことをはっきり悪いというのはかなりむずかしいと思われますが，少なくともいいものはいいという場合には説得力を持たせてほしいと思います。特に今回取り上げているデザインという面については，客観的に評価するのはきわめてむずかしいものですから。

本誌のバックナンバーに載った記事をざっと見ることにしました。その結果，国産のマシンとしてアフォーダンスを述べる対象となりうる，数少ないマシンのひとつ，X68000のデザインに言及した本誌の記事（文献２）を見つけることができました。

その記事はX68000本体の色（グレー）の持つ現代性や，高層ツインビルのようなマンハッタンシェイプのかっこよさを指摘するとともに，取っ手を出したときのイメージを「出前のおかもちの世界」と形容して，その親しみやすさを読者に知らせています。X68000の持つアフォーダンスのフォーマル性とカジュアル性が明確に記述されているように感じられました。

川崎氏の主張しているデザインの持つ役割，あるいはデザイナーの義務というものは，実はすべての人のそれぞれの持ち分において，まったく同じような構造が見い出されるようなきわめて本質的な問題であると考えられます。

僕自身，川崎氏の記事によって，自分の仕事における同様な問題構造を頭の中に定位させられました。そのことについては，また機会があれば述べさせてもらえばと思っています。

ところで，今回紹介した川崎氏の連載記事の中には，X68000は一度も登場していません（Macintosh専門誌ですからね）。川崎氏はX68000の持つアフォーダンスに対して，どのように感じたかを聞いてみたいものですね。

参考文献

1）川崎和男，DesignTalk：Macintosh Interaction 9，マックパワー 1991年5月号，pp.168-170
2）祝一平，X68000ショッキングデビュー，Oh! MZ 1986年11月号，pp.23-25

PENGUIN INFORMATION CORNER

ペ・ン・ギ・ン・情・報・コ・ー・ナ・ー

NEW PRODUCTS

「書院」新2機種
WD-SD70/A810
シャープ

WD-SD70

シャープは縦横自在の液晶大画面のプロフェッショナルユーザー向けの書院「WD-SD70」と，省スペース型でオフィス向けの書院「WD-A810」の2機種を発売した。

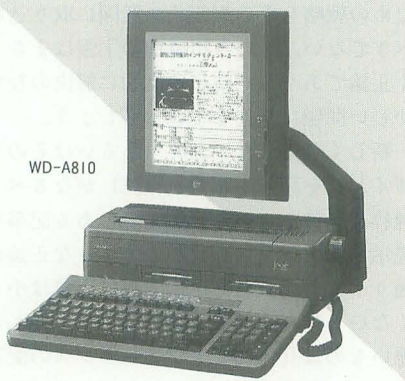
「WD-SD70」は業界最大で高精細の液晶画面を持ち，さらにそれを縦横に回転させ，文書を一覧して入力することができる。液晶画面の大きさは17インチCRTに相当し，約123万画素（960×1280ドット）の高精細タイプを採用。縦置きの場合は24ドット文字表示でA4縦文書の1ページ，横置きの場合では16ドット文字表示でほぼB4横，あるいは12ドット文字表示でA3横文書の1ページ全体を一覧しながら入力/編集することができる。

このほか，40Mバイトハードディスクの搭載，400DPI(ドット/インチ)レーザープリンタ対応などにより，CRT画面の本格ビジネスワープロに匹敵する高度な機能と操作性を持っている。しかし，液晶画面採用により，占有面積はこのような本格ビジネスワープロと比べて1/2という省スペースを実現している。

「WD-SD70」はプロユーザーのためのワープロということになっているが，もうひとつの「WD-A810」はビジネスなどの場における パーソナルワープロという位置付けになっている。液晶画面は約51万画素（640×800ドット）の高密度で，A4フルページの表示が可能。また，液晶表示部分と本体をアームで接続する新機構の採用により，置き場所や照明の具合などに合わせて画面をセッティングすることもできる。

「WD-A810」も本体の奥行き18.3cmという省スペース設計になっている。また，「WD-A810」の姉妹機として，レーザープリンタ用の書式対応の「WD-SD50」も発売される。「WD-SD50」は直接レーザープリンタに出力することはできないが，レーザープリンタで印刷することを前提に文書を作成したり，レーザープリンタ用に作成された文書を編集/修正することが可能。

価格は「WD-SD70」が950,000円，「WD-A810」が348,000円，「WD-SD50」が398,000円となっている。また，省スペース型のB4レーザープリンタ「WD-02LP」もあわせて発売された。こちらの価格は450,000円（価格はすべて税別）。

〈問い合わせ先〉
シャープ㈱　☎03(3260)1161，06(621)1221

WD-A810

"写嬢"シリーズ最上位機種
FR-3000
日本アビオニクス

日本アビオニクスは，フィルムレコーダ"写嬢（シャガール）"シリーズの最上位機種として「FR-3000」を発売した。

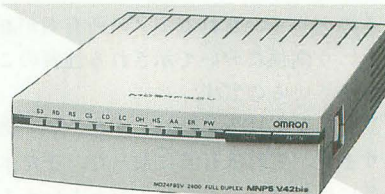
「FR-3000」は高画質コンピュータグラフィックなどに最適なアナログRGB方式のマルチスキャンフィルムレコーダである。64kHz以上の水平走査周波数，高画質のワークステーションに対応するため，40～80kHz帯でのマルチスキャン方式を採用している。ワークステーションなどで制作される各種のCG画像を短時間で最終成果としてのフィルムの形で出力でき，印刷用の版下として使用，あるいはポスター大の拡大まで十分に耐える高画質を実現している。使用できるフィルムサイズは35mm，4×5，8×10(インチ)のほか，インスタントフィルムなど，27種類と豊富になっている。

価格は980,000円(税別)。

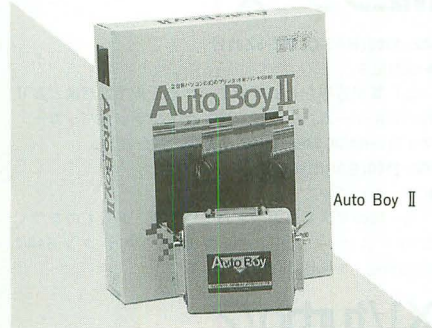
〈問い合わせ先〉
日本アビオニクス㈱　☎03(3725)7814

FR-3000
CCITT V.42bis搭載コンパクトモデム
MD24FB5V
オムロン

オムロンは，「MD24FS5」の完全上位互換機種で，あらたに国際標準データ圧縮機能としてCCITT（国際電信電話諮問委員会）が勧告したCCITT V.42bisを標準搭載した「MD24FB5V」を発売した。

「MD24FB5V」は「MD24FS5」の完全上位互換機種でありながら，体積比約1/4とコンパクトなボックス型モデムである。データ圧縮機能にCCITT V.42bisを標準搭載しているので，ソフトなどの工夫なしで実効通信速度が最高約3倍となる。また，MNPクラス5も標準搭載しているので，

MD24FB5V

MNPの通信環境でも最高約2倍の実効通信速度となる。価格は39,800円（税別）。
〈問い合わせ先〉
オムロン㈱　☎03(5488)3219

低価格自動プリンタ切り替え器
Auto Boy II
八戸ファームウェアシステム

Auto Boy II

八戸ファームウェアシステムでは自動プリンタ切り替え器「Auto BoyII」を発売した。この「Auto BoyII」は「Auto Boy」の廉価版にあたり，ケーブル込みの価格では最も低価格な自動プリンタ切り替え器である。

「Auto BoyII」ではパソコン2台対プリンタ1台をつなぐことができる。機能は従来どおり，どのパソコンのデータを印字するのかを自動的に判別，瞬時に切り替え，印字終了後は自動的に次のパソコンの印字を開始するようになっている。

価格は12,800円（税別）。
〈問い合わせ先〉
八戸ファームウェアシステム㈱
☎011(716)3815

INFORMATION

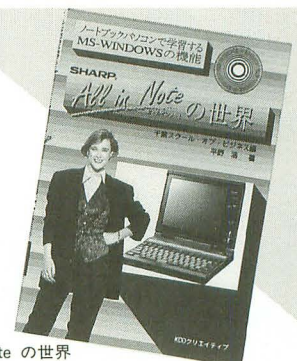
アイデアコンテスト
「未来派コンピュータ」
オムロン

オムロンは，UNIXワークステーション「LUNA-II」の発売を記念して，未来のコンピュータについて自由なアイデアを募集するアイデアコンテスト「未来派コンピュータ」を次の要領で実施する。
○テーマ
未来のコンピュータ
・いまのコンピュータをこのように改良したい
・こんな仕様のコンピュータなら使ってみたい
・こんな分野でコンピュータを使ってみたい
・2001年にはコンピュータはこうなっている
など，いろんな角度から自由に。
○応募形式
文章，イラスト，音声のいずれか。ひとりで何点でも応募できるが，1回の応募につき1アイデアのこと。
・文章　形式自由
・イラスト　簡単な説明をつける
・音声　テープに録音
○応募条件
応募アイデアは未発表，かつ応募者自身のアイデアによるものに限る。
○賞品
未来派コンピュータ大賞（10名）
図書券10万円分
LUNA賞（1000名）
LUNAグッズセット（オリジナルテレホンカード，Tシャツ，ステーショナリー）
○応募の締め切り
9月30日（当日消印有効）
○応募先
住所，氏名，年齢，電話番号，職業を明記のうえ，下記へ。
〒140　東京都品川区北品川4-7-35御殿山森ビル13F　オムロン株式会社
「未来派コンピュータ」アイデアコンテスト係
〈問い合わせ先〉
オムロン㈱　☎03(5488)3253

JACA
'91日本イラストレーション展
国際芸術文化振興会

JACA日本イラストレーション展はイラストレーションの発展と新人育成を目的に，1983年より毎年開催している。また昨年度より，公募内容を刷新，「テクニカル/デジタル部門」が増設された。
○応募部門
A部門　フリー・イラストレーション
自分自身の発想による自由な表現としてのイラストレーション
B部門　テクニカル・イラストレーション
精密で技術的な表現を重視したイラストレーション
C部門　デジタル・イラストレーション
CGなど新しいテクノロジー，メディアに対応したイラストレーション
○応募規定
オリジナル作品で未発表作品に限る。出品点数の制限はない。
大きさは最大B0（1030×1456mm）～最小B3（364×515mm）サイズ。
○応募料金（各部門共通）
ひとり3点まで，8,000円。3点以上の場合は追加1点につき1,000円。
○本年度審査員
粟津　潔，安西水丸，河本信治，田中一光，中篠正義，南条史生，ダン・ファーン
○応募の締め切り
直接搬入　6月1日（土），2日（日）
輸送応募　5月20日（火）
○賞
大賞　1名 100万円
（副賞：海外展会場までの往復航空券）
金賞　1名 100万円，銀賞　2名 50万円，
銅賞　5名 10万円
○1991年度展覧会
東京展　伊勢丹美術館　1991年8月
その他国内数カ所および海外で開催
〈問い合わせ先〉
社団法人　国際芸術文化振興会
☎03(3582)0631

BOOK

All in Noteの世界
KDDクリエイティブ

All in Note の世界

シャープから発売されている，ノート型AX仕様パソコン「All in Note」は，20Mバイトハードディスク内蔵，そして，MS-WINDOWS ver.2.11およびWINDOWSベースの統合型ソフト「ビジネスメイト」を標準装備している。

本書ではこの「All in Note」の機能を，これからのMS-DOSマシンの標準的な利用環境になると思われる，MS-WINDOWSを中心に解説されている。

何もわからない人が最初のページから学習すれば，ページが進むにつれてわかってきて，最後まで学習すると使いこなせるようになっている。本書はこのような解説書を目指して作られており，8つの演習を用意してより実践的に学習できるようになっている。
〈問い合わせ先〉
㈱ウッドブック　☎03(3207)3697

FILES Oh!X

このインデックスは，タイトル，注記——筆者名，誌名，月号，ページで構成されています。ゴールデンウィークも終わり，やっといつもの生活に。花粉症の人，つらいでしょうが，あともう少しの辛抱です。

一般

▶NETWORK CONNECTION
AMIGA専門のネット「Orange Agnus」の紹介やNIFTY-Serveの「ぴあCD comming soon」サービスなど。PDSはX68000のシューティングゲーム「RED ARMS」。——編集部，LOGIN，7号，262-265pp.

▶HOT INFORMATION
女性にも使えるファッショナブルなデザインのシャープのスタイリッシュ電子システム手帳「PA-X1」を紹介。——編集部，マイコンBASIC Magazine，5月号，96p.

▶どこでもゆくぞ日本パソコン百景
大阪パソコンの陣の巻と題して日本橋にオープンした上新，ニノミヤの新店舗を取材。両店ともにハイセンスな店づくりと新傾向のサービスが自慢だ。——フデヨシ＆カシワラ，ASCII，5月号，222-223pp.

▶もうパソコン通信しかない
今すぐパソコン通信を始めたい人のためのガイド。始めるまでのステップやパソコン通信のメリット，BBSとVANの選び方，Communication PRO-68k Ver.2といったソフトやモデムのガイドなどなど。——編集部，ASCII，5月号，225-248pp.

▶MEDIA BREAK
新宿駅西口地下広場に登場したAVからくり時計をレポート。40台のモニタを使った時計はなんとX68000によって管理されている。そのほか秋葉原の開発基本計画，アマチュアCGAコンテストの結果紹介など。——編集部，ASCII，5月号，377-379pp.

▶東京新宿・新都庁舎
東京の副都心，新宿の摩天楼にあらたに加わった東京都庁新庁舎。日本を代表するこのインテリジェントビルの素顔に迫り，その体内の電子回路をのぞく。——野沢潤一郎，マイコン，5月号，220-223pp.

▶入門ハード工作室
ロジック回路の基本を目で見えるようにするボードを作る。さらにAND，NAND，OR，NOR，INVERTER(NOT)などのゲート操作の基本について解説する。——石川至知，マイコン，5月号，308-314pp.

▶C言語事始め
C習得の壁を乗り越えるための体験談，12個の必須関数の徹底解説，今話題の“C＋＋”入門編などの記事とCコンパイラのカタログ。——吉沢正敏・山崎仁史・知見光泰，I/O，5月号，73-115pp.

▶DB-Zを徹底的に使いこなす
シャープ電子手帳PA-9500についての解説。今月は名刺管理機能について触れる。今までの電話帳機能との違い，使い分けのポイントなどを説明。——松田ぱこん，ポケコンジャーナル，5月号，56-58pp.

MZシリーズ

MZ-1500(BASIC MZ-5Z001)

▶COMALS
たし算が苦手だとムズいぞ。タイトルでご想像どおりのパズルゲーム。——テンチング・フラットバッカー，マイコンBASIC Magazine，5月号，120-121pp.

MZ-2500(BASIC-M25)

▶ロールパズル
16×16のマス目に表示されている形と同じものをつくるパズルゲーム。——坂井重典，マイコンBASIC Magazine，5月号，122-124pp.

X1/turbo/Z

X1シリーズ

▶DEVIL CARD
目の前にあるカードを，降ってくるカードめがけて打つ。打ったカードと当たったカードが同じ種類なら手持ちのカードとなる，アクションパズルゲーム。——UGN SOFT，マイコンBASIC Magazine，5月号，147-149pp.

▶ランダムRPGニルナ・ノーグ
リスト短め。X1版ティルナ・ノーグ(？)。——松原拓也，マイコンBASIC Magazine，5月号，150-152pp.

X1＋FM音源ボード（要NEW FM音源ドライバ）

▶Xak～オープニング・テーマ～
マイクロキャビンのゲーム「Xak」のオープニングテーマのミュージックプログラム。——大倉章，マイコンBASIC Magazine，5月号，181-183pp.

X1turboシリーズ

▶がむしゃらパンチキッド
スクリーンを4枚使ったハイスピードアニメーション処理のスーパーボクシングゲーム。——堀田英克，マイコンBASIC Magazine，5月号，153-154pp.

X68000

▶NEW SOFT
発売予定の「パロディウスだ！」「ファランクス」「石道」を紹介。——編集部，LOGIN，7号，12-24pp.

▶X68000新聞
発売予定のゲーム「スコルピウス」や，既発売の「Ryu～哭きの竜～より」の紹介。X68000 4年間，システムの歴史など。——編集部，LOGIN，7号，242-245pp.

▶NEW SOFT
スコルピウス，サブナック，ビーストロード，ファランクスを紹介。——編集部，LOGIN，8号，14-20pp.

▶THE NEWS FILE

参考文献

I/O　工学社
ASCII　アスキー
コンプティーク　角川書店
C MAGAZINE　ソフトバンク
テクノポリス　徳間書店
ポケコンジャーナル　工学社
POPCOM　小学館
マイコン　電波新聞社
マイコンBASIC Magazine　電波新聞社
LOGIN　アスキー

新刊書案内

BMCというのは，放送音楽文化振興会のこと。浅田彰氏は経済学者だが，一般には思想的な受け止められ方をしている。NHKっぽい名前のBMCと浅田彰は一見ミスマッチだが，BMCが放送メディアの将来像を求めており，浅田彰氏がハイビジョンやメディアアートなどに関わっていることを考えると，実にいい組み合わせだ。

今，ハイパーメディア＝退屈な百科事典，ハイビジョン＝NHK的映像，というイメージがある。それはよくない。浅田彰氏はそれに危惧を抱き，ハイ・イメージに相応しいイマジネイションが問われている状況といっている。本書は，そういう観点から，ハイビジョンに限らないさまざまなメディアの未来を見据えた18人の記事や対談で成り立っている本である。CGでお馴染みの原田大三郎や立花ハジメ，映像作家のリプチンスキーなどそうそうたるメンバー。ややこしいいい回しが多くてわかりにくい文章もあるが，将来のメディア（特に映像や放送）について，最前線の人々が何を考え，何をしようとしているかを知るにも，一度読んでみるべきだろう。（K）

ハイ・イメージ・ストラテジー　浅田彰監修　BMCイメージ・プロセッシング研究会編　福武書店刊
☎03(3230)2131　A5版　285ページ　2,000円

X68000の新機種，高速16MHzMC68000搭載のX68000XVIを紹介している。──編集部，LOGIN，8号，30p.
▶最新ゲーム徹底解剖!!
続ダンジョン・マスター カオスの逆襲と，発売予定のノスタルジア，マーブル・マッドネスの紹介＆攻略。──編集部，LOGIN，8号，152-165pp.
▶X68000新聞
颯爽と登場X68000XVI！ ユーザー待望の高速版X68000がついに登場。特集を組んで，そのスペックを紹介している。新着ゲームは「遥かなるオーガスタ」「ランペルール」「ファンタジーⅣ」「キャンペーン版大戦略Ⅱ」など。PDSはMIDI演奏システム「RC」を紹介している。──編集部，LOGIN，8号，224-231pp.
▶GAMING WORLD
マーブル・マッドネス，サブナック，シグナトリー，パロディウスだ！，スコルピウスなどを紹介。──編集部，テクノポリス，5月号，20-34pp.
▶SOFT EXPRESS
メルヘンメイズ，シグナトリー，シムシティー・テレインエディター，遥かなるオーガスタや，発売予定のスコルピウス，ファランクス，サブナックなど。同人ソフトはゲームのSTOX，PARORANを紹介。──編集部，コンプティーク，5月号，80-89pp.
▶How To Win
続ダンジョン・マスター カオスの逆襲を攻略。──編集部，コンプティーク，5月号，136-139pp.
▶電脳通信WHAT'S NEW
X68000の新機種「X68000XVI」を紹介。──編集部，コンプティーク，5月号，252p.
▶新製品紹介
新機種X68000XVIを紹介。そのスペックや，バージョンアップされた付属ソフトなど。──北澤充裕，マイコンBASIC Magazine，5月号，67-71pp.
▶誌上公開質問状
BASICを立ち上げたときに，ファンクションキーを好きな状態にしておくには？ 立ち上げたときに，辞書学習機能を「ディスク学習」にしておくには？ など。──多田太郎，マイコンBASIC Magazine，5月号，91-92pp.
▶FUUSUKE
フースケ（主人公）を操作してシャボン玉を落とさずゴールまで行く。──高橋潤，マイコンBASIC Magazine，5月号，155─157pp.
▶ボールウォーズ
重力制御装置の戦い。2人用対戦ゲーム。──少年1号，マイコンBASIC Magazine，5月号，158─159pp.
▶Final Star
最終兵器「スーパートロッコ」で地球のために戦う1人の少年。──平田慎之助，マイコンBASIC Magazine，5月号，160-162pp.
▶アーケード版 パロディウスだ！ ～アイランド・オブ・パイレーツ～
コナミのゲームミュージックプログラム。要NAGDRV＋CM-64。──野口英二，マイコンBASIC Magazine，5月号，184-189pp.
▶MD版ソーサリアン「ツインアイランズ」
日本ファルコムのゲームミュージックプログラム。要NAGDRV＋MT-32系MIDI音源。──渡辺和広，マイコンBASIC Magazine，5月号，190-193pp.
▶今月の注目ソフト
オリジナルシューティングゲーム「スコルピウス」と，対戦ゲームもたのしいアクションゲーム「ボンバーマン」を紹介。──鹿島五郎・石井弘一，マイコンBASIC Magazine，5月号，244-245pp.
▶Hot Press
サブナック，遥かなるオーガスタ，ノスタルジア，キャンペーン版大戦略Ⅱ，パロディウスだ！，ファランクス，スコルピウス，A列車で行こうⅢなど。スピードアップした人気のX68000XVIを紹介。──編集部，POPCOM，5月号，17-26pp.
▶ゲームの達人
マーブル・マッドネスと続ダンジョン・マスター カオスの逆襲。──編集部，POPCOM，5月号，88-92pp.
▶ミュージック・パビリオン
大ヒットしたKANの「愛は勝つ」のミュージックプログラム。──編集部，POPCOM，5月号，159-162pp.
▶AV STRASSE
新デザインの16MHzバージョンX68000XVI。その概観や変更点，高速化の効果などについて述べる。それからPDSのリアルタイムメモリダンププログラム，memscope.xを紹介。──編集部，ASCII，5月号，353-356pp.
▶MiCRO MUSiQUES
MIDI音源の音色の番号を統一しようというローランド提唱のGS規格とその製品第1弾となるSoundCanvasを紹介する。ほかにX68000，PC-9801，MIDIの間のコンバート環境を図示。──編集部，ASCII，5月号，357-360pp.
▶新ハード徹底研究
X68000シリーズの最上位機種として発表されたX68000XVIについて内容を紹介し，ソフトウェアの変更点，ロードテスト，市販ソフトでの高速性のテスト，ハードウェアについての総括，今後の展望などを掲載している。──高橋雄一，マイコン，5月号，111-122pp.
▶MYCOM SOFT REVIEW
先頃バージョンアップしたMusicstudio PRO-68K Ver.2.0を取り上げ，内容と完成度について批評する。やはりMIDI＋OPM＋PCM個別コントロールは強力なようだ。──都築敏也，マイコン，5月号，156-159pp.
▶なんでもQ＆A
プリンタで縦書きをきれいにするには？ キー設定ファイルの内容を変更して定着させるには？ CARD PRO-68Kの概要について。──シャープ液晶映像システム事業部第2商品企画部，マイコン，5月号，380-381pp.
▶NEW Machine
X68000XVI登場のニュース。16MHzの効果などについて述べる。──牛島健雄，I/O，5月号，126-127pp.
▶FM音源用音色エディタさんでぃ
SOUND PRO-68Kを買わなくても音色は作れる！ というわけで，マウスオペレートのFM音源エディタ。SOUND PRO-68K形式のデータ読み書きも可能。──橋口湖，I/O，5月号，142-148pp.
▶SRAMDISK.SYS
SRAMの容量を1Kバイトアップするためのユーティリティ。ウイルス防止の効果もあるが，多少のリスクもあるので注意。──（は），I/O，5月号，149-153pp.
▶AutoRetract
ハードディスクのクラッシュ防止用，自動ヘッド退避プログラム。Human68k Ver.2.0以上が対象。──利器男，I/O，5月号，154-156pp.
▶GNU奮戦記
GCC Ver.2の概要について述べる。付録ディスクにXGCCとライブラリを収録。XGCCのソースファイルなども6月号以降に収録する予定である。──吉野智興，C MAGAZINE，5月号，144-146pp.

ポケコン

PC-E500
▶TEN-TRIS 2
右端から降ってくる数字で，合計がちょうど10になるようにしてすべての数字を消しましょう。──西村論，マイコンBASIC Magazine，5月号，164─165pp.
▶Nagashino!
配られてくるカードで戦う，戦国シミュレーションゲーム。──中村啓之，マイコンBASIC Magazine，5月号，166-167pp.
PC-E500/E550/1480U/1490U
▶BASIC入門コース
初心者を対象にしたポケコンの操作法，BASICプログラミングの初歩的知識の解説。──塚田洋一，ポケコンジャーナル，5月号，4-16pp.
▶最後の刺客
忍者モノのアクションゲーム。同族を皆殺しにした鬼一族を倒せ！──きゃっちゃん，ポケコンジャーナル，5月号，67-77pp.
▶残照街道
以前に発表された凶刃乱舞の番外編。行く手をふさぐ刺客を斬り伏せて，時間内に目的地までにたどり着け！──双剣士，ポケコンジャーナル，5月号，78-84pp.

メカノ
どう表現したらいいかわからないが，「美学と，科学と哲学の，相互参照的な三角測量法を……」だそうである。つまり，機械を哲学するというか，そういうものだ。理科系側からのコンピュータを通した美や哲学へのアプローチは，我々にとってより興味深い。機械と人間の関係を探るうえにおいても，避けられない視点だろう。一見，わけのわからない妖しい本だが，その妖しさが重要だし，気持ちいいのだ，と思う。 (K)
杉田敦著 青弓社刊
☎03(3265)8548 四六版272ページ 2,060円

科学なるほど事典
前書きから引用しよう。「うんちくを傾けるなら，なにも食文化ばかりではありません。現代のような文明社会に生きているのですもの，ひとつ，ベーシックな科学ネタで差をつけてみませんか？」というわけで，そういう本である。内容は実に平易で，レベルも低いので，ちょっとした読み物といった風情だ。うんちくなどという言葉が前書きで出てくることでもわかるとおり，いつでも読める雑学集と思えばいい。 (K)
鍵谷健司・宇井克人著 ナツメ社刊
☎03(3291)1257 B6版 223ページ 1,000円

QUESTION and ANSWER

市販のシューティングゲームでBGがスクロールすると，どんどんBGが変わっているのですけども，どうやっているのですか？ BG_SCROLLを使うと同じ背景がループしてしまいます。機種はX68000 EXPERTⅡです。X-BASICでお願いします。

秋田県　青木　学

市販ゲームでのBGの使われ方を見ると，ほとんどが256×256の画面モードでBGを2ページ使っているようです。

一般的にいって，シューティングゲームなら，BG0に得点表示や自機の残機表示，BG1に背景を表示してスクロールといった使い方が考えられます。また，R-TYPEなどにもある巨大戦艦などの大きなキャラクタは，BGに固定部分（戦艦の胴体部分）を表示して，可動部分（砲台）をスプライトで表示，といったこともありえます（調べたわけではないので，推測の域を出ませんが）。

さて，X68000で扱えるBGの大きさは縦64個×横64個となっていて，設定できるパターンの大きさは表示画面に応じて，

表示画面	パターン	
512×512	16×16	(1024×1024)
256×256	8×8	(512×512)

と決められています(カッコ内はBGの大きさ)。これからもわかるように，いずれの場合も一度に画面に表示できるパターンの最大数は，32×32＝1024個で，BG画面全体の1/4となっています。

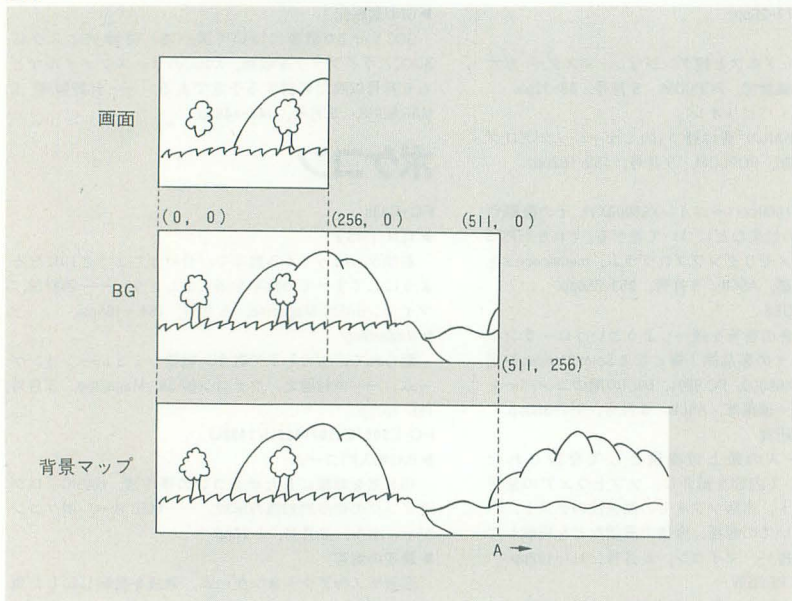
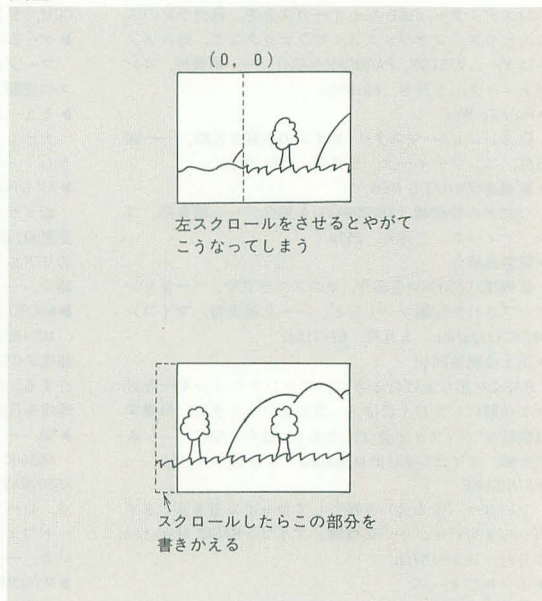
話を戻して，BGを繰り返し表示させない方法を考えてみます。図1に背景とBGと表示画面の関係を示します。この状態からBGを左にスクロールしていくと，やがて画面には山全体が現れ，海になり，山のはじっこが出たところで，最初の草木が再び表示されます(図2)。これを見て青木さんは「同じ背景が何回もループしてしまう」といっているのでしょう。

図2をよく見てもらうと，上に(0,0)とありますが，これがBGの座標を表していることを理解してください。どうして背景がル

図1

図2

左スクロールをさせるとやがてこうなってしまう

スクロールしたらこの部分を書きかえる

リスト1

```
 10 screen 1,2,1,1
 20 dim char font(255)
 30 int i,j,k,count=0
 40 sp_init()  /* スプライト、BG初期化
 50 sp_disp(1)  /* スプライト、BG表示
 60 font_set()  /* パターン設定
 70 sp_color(1,65535,1)  /* パレットブロック設定
 80 sp_color(2,36000,1)  /* パレットブロック設定
 90 bg_set(0,0,1)  /* テキストページ0をBG0に設定
100 /*
110 /* BG0(256,0)-(511,255)に
120 /* ランダムにパターンを設定する
130 /*
140 for l=32 to 63
150     bg_put(0,l,int(rnd()*32),&H141)
160 next
170 /*
180 j=1
190 i=(i+4) mod 1024   /* iが1024を超えないようにする
200 bg_scroll(0,i,0)   /* BGを4ドットスクロールさせる
210 count=count+1      /* スクロールカウンタを増やす
220 if count=4 then {  /* 4ドット×4回=16ドット
230                    /* スクロールしたらBGを書き換える
240   k=bg_stat(0,0)/16-1
250   if k<0 then {
260     k=63
270     j=(j+1) mod 26 }
280   for l=0 to 31
290       bg_put(0,k,l,&H120)
300   next
310   bg_put(0,k,int(rnd()*32),&H141+j)
320   count=0 }
330 goto 190
340 end
350 /* フォント設定
360 func font_set()
370 int i
380 sp_def(&H20,font)
390 for i=&H41 to &H5A
400   symbol(2,1,chr$(i),2,1,1,2,0)
410   symbol(0,0,chr$(i),2,1,1,1,0)
420   get(0,0,15,15,font)
430   sp_def(i,font)
440   fill(0,0,19,19,0)
450 next
460 endfunc
```

ープしてしまうかというと，一度画面から消えたBGが再び表示されてしまうからです。これを防ぐには，画面左から消えていったBGに，図2A点以降の背景を設定しておけばいいことがわかると思います。

BG座標(511,0)の次が（0，0）だなんておかしいと思う人もいるかもしれませんが，BGが左端と右端（上端と下端）がつながっている球面状の構造をしている，ということを知っておいてください（これはグラフィック画面でも同じ）。つまり，BG_SCROLL(0,384,0)としたときには，画面左半分にBG0の(384,0)－(511,255)が，画面右半分にBG0の(0,0)－(127,0)が表示されるということです。

とりあえずサンプルプログラムを作成しました。プログラムを実行すると，画面にアルファベットが表示され，横スクロールしていきます。このプログラムでは表示画面を511×511に設定しているので，BGに設定できるパターンは16×16です。ですから16ドットスクロールするたびに，画面から消えたBG部分を書き換えるようにしています（240～320行）。

シューティングゲームだったら，当然マップデータがあるでしょうから，それを参照してパターンをセットするようにすればいいでしょう。

なお，X-BASICで書いたために，スクロールが多少ガタガタしていますが，コンパイラにかけるとか，最初からアセンブラで書いてしまえば問題ありません。しかし，逆にスクロールが速くなりすぎると，今度は垂直帰線期間中にスクロールさせるように改良しないと画像がぶれてしまうので注意してください（4月号の質問箱を参考にしてください）。

XC ver.2.0のMAKEについて聞きたいのですが，ソースファイルを“A:¥SRC”のディレクトリ，オブジェクトファイルを“A:¥OBJ”にしてMAKEをしたいのですがうまくいきません。オブジェクトファイルはちゃんとA:¥OBJにできるのですが，未変更のソースが毎回コンパイルされてしまいます。MAKEのオプションは，

MAKE /f MK

です。A:¥SRCにオブジェクトファイルがあればそんなことはないのですが。どこがまずいのですか。　大阪府　真本　順景

プログラムを開発する人にとって，MAKEのある環境は大変に快適なものです。MAKEを知らない方のために，ひと言で説明するなら「複数のオブジェクトファイルからなる実行ファイルを作成する際に，アップデートされたソースファイルを自動判別してアセンブル，またはコンパイルして，目的の実行ファイルを自動作成するツール」となります。なーんだ，それならバッチファイルで十分じゃないかと思われる方がいらっしゃるかもしれませんが，バッチファイルの場合，10数個のソースのうち，たった1個について更新したときも，すべてのソースファイルがアセンブル，コンパイルされてしまいます（MAKEは未更新のソースファイルをアセンブル，コンパイルしない）。

MAKEが必要とする実行ファイルの作成に必要なソースファイルや，オブジェクトファイルのあるディレクトリなどの情報を記述しておくファイルが“メイクファイル”と呼ばれるものです。MAKEでは，さまざまなマクロ機能がサポートされているので，これを利用することによって簡潔にメイクファイルを記述することができます。反面，マクロを使ったメイクファイルを初めて見る人にとっては，なにをするものなのかまったくわからないでしょう。

前置きが長くなりましたが，真本さんのメイクファイルをリスト2に紹介します。ファイル名が“MK”となっているので，MAKEの/fオプションを使ってメイクファイルを指定しています。質問の内容は，未変更のソースが毎回コンパイルされてしまうというのですから，事態は深刻です。

質問にもあるように，A:¥SRCにオブジェクトファイルがあればうまくいくようですが，オブジェクトファイルのあるディレクトリ指定に誤りがあるのでしょう。リスト2では5，11行でオブジェクトファイルを指定しています。真本さんはA:¥SRCからMAKEを使うようですから，MAKEは5，11行のオブジェクトファイルを，A:¥SRCのなかから検索しています。4行あたりに，

OBJDIR = A:¥OBJ

と挿入して，5行を，

OBJS = $(OBJDIR)¥C_MAIN.o ……

と，すべてのファイル名にそれぞれ$(OBJDIR)をつけ，11行も，

$(OBJDIR)¥%.o: %.c

と変更すればうまくいくと思います。

（影山裕昭）

リスト2

```
 1: CC = CC
 2: CFLAGS = /Ns /Fc /A3 /Y /W
 3: PROG=TEST.X
 4:
 5: OBJS = C_MAIN.o CPIC.o C_FILE.o C_INIT.o C_VERI.o ¥
 6:         C_GLIB68.o C_CRT.o
 7:
 8: $(PROG): $(OBJS)
 9:         lk /i SR_LNK
10:
11: %.o: %.c
12:         $(CC) $(CFLAGS) /FoA:¥OBJ¥$(@F) $< >ERR
13:
14:
15: *注  インダレクトファイル SR_LNK は省略します
16:
```

質問にお答えします

日ごろ疑問に思っていること，どんなことでも結構です。どんどんお便りください。難問，奇問，編集室が総力を上げてお答えいたします。ただし，お寄せいただいているものの中には，マニュアルを読めばすぐに回答が得られるようなものも多々あります。最低限，マニュアルは熟読しておきましょう。質問はなるべく具体的に機種名，システム構成，必要なら図も入れてこと細かに書いてください。また，返信用切手同封の質問をよく受けますが，原則として，質問には本誌上でお答えすることになっていますのでご了承ください。なお，質問の内容について，直接問い合わせることもありますので，電話番号も明記してくださいね。

宛先：〒108 東京都港区高輪2-19-13
NS高輪ビル
ソフトバンク株式会社出版部
「Oh! X質問箱」係

FROM READERS TO THE EDITOR

そろそろ，環境が変わったことによる，あわただしさもいち段落している頃。新歓コンパや花見で疲れた肝臓も，ほっとひと息ついているかな。これからおとずれる，うっとうしい梅雨をのりきるためにしっかり体力をつけておきましょうね。

◆4月号の特集を読んでいて思ったんですけど，見ただけで心がときめくゲームが少なくなったような気がします。特に，ARPGにです。このてのゲームは，見た感じでだいたいの目安といいますか，楽しさが想像できると思うのですが，この頃のARPGはみんな似ていて，なにか光るものがなくなってしまったような気がします。

菅沼　光剛(17)埼玉県

すべてメーカーまかせではなく，ユーザー自身もしっかりしなくてはなりませんね。

◆ゼビウススティック，懐かしいですね。わざわざゼビウスを2つ買って，エグゾアII，ウォーロイド，マリオブラザーズSPECIAL，パンチボールなどをするときに役立てたものでした。いまではひとつになったけれど，まだまだワースタやワールドコートをするのに使っていますが，やっぱりこのスティックが使いやすいですね。

近藤　英二(19)愛知県

まともに動作するものならいいですが，編集部にあるやつは使いすぎで手を離しても動きつづけるんですよ。

◆ダンジョン・マスターの推薦理由を見てたら，「モンスターが，突然出てきて心臓が飛び出そうになったのはこのゲームだけだ」とある。ふふん，君はROGUEをやったことがないのだな。POTION OF PARALYSISもWAND OF SLOW MONSTERも使い尽くしてビクビクしながら暗闇を歩き，目の前に突如“G”が現れたときの，全身に電気が走るような感じは，とても言葉では説明できない。ROGUEは凄いゲームだ（いまだに生還できない）。　石田　伯仁(17)神奈川県

ただのアルファベットに，あれほど入れ込むゲームもなかなかありませんね。

◆ああ，まだ落ち着かない。生まれて初めて目撃してしまった。それは，夜10時過ぎに帰り道を急いでいると，かっとばしたK自動車が私の横を通りすぎたかと思うと，交差点でドカン。Kはそのまま50メートル以上もふっとび，もう1台のぶつかった乗用車もふっとんで，くるっと回って電柱に激突。どっちもメチャクチャ。さいわい怪我ですんだみたいで，急いで家に帰って親にその話をするとひと言，「お前はどこでそれを見ていたの？」そうなんです。よく考えると私はその現場から5メートルと離れていなかったのです。だから，もしもあのKがこっちにふっとんでいたら……。本当，自分も気をつけなければ。　小宮　崇(19)埼玉県

あぶなかったですね。乗っていた人もたいしたことがなければいいのですが。

◆X68000も最近ではゲームとワープロにしか使わなくなってしまった。アテにしていたバイトが入らず，2カ月もの長い春休みを持て余している状態だ。スキーも2回行ったところで金が尽きた。21年間も人間やっていれば，たまにはこんな停滞期に入ってしまうこともあるのだろうか（しかし，カビが生えてきそうだな，こりゃ）。

小藪　賢(21)埼玉県

社会人になれば嫌でも忙しくなるし，いまのうちに暇を満喫しといてもいいんじゃない？

◆あー，Cコンパイラがほしい。MIDIボードと音源がほしい。しかし，この3～4月は地獄のゲーム発売攻撃なのである。「メルヘンメイズ」「マーブルマッドネス」「キャンペーン版大戦略II」「遥かなるオーガスタ」「パロディウスだ！」「A列車で行こうIII」……。6月になれば，ボーナスが入るからそれまで待つか。

信太　徹(21)神奈川県

ソフトは買えても遊ぶ暇がなかったりして。

◆X68000ACE-HDが我が家に来てはや2週間。ようやく環境も整い，プログラムを作ろうかな，というところ。でも，いざ作ろうと思うと今度は何を作ろうか迷ってしまった。ぼーっとしててもしかたないので，ショートプロぱーてぃのrmazeとwakを打ち込んだ。わー，rmazeの速くて気持ち悪いこと。wakはちょっと遅いかな。でも，きれい。そうだ，しばらくはこんな短いプログラムを入力して勉強しよう。そう思ったしだいです。　西田　文彦(20)神奈川県

ま，気長にやりましょう。

◆1年半ぶりに箱の中からMZ-2200を出した。昔のソフトって飽きませんね。画面も音楽も地味なのに，はまってしまうのだ。それにしても「北斗の男」は名作だなあ，と思う夜。

百瀬　孝彦(21)神奈川県

わけもなく昔のソフトを引っ張りだして，思い出に浸る。ちょっと年寄りくさいかな。

◆春から短大生です。1浪して，予備校も行かずに宅浪して，アルバイトをしまくり，車の免許を取り（9月），1月のセンター試験に失敗し，それにもめげず短大だけど合格しました。さあ，これから勉強して，4年制のほうへ入るぞ。でも，自分は農学部へ行きたかったけど，ここは電子工学部。まあいいか。X68000もXCもあるし，XBAStoCもあるからがんばろ。

小篤　克典(19)大阪府

せっかく立てた目標ですから，どっちつかずの状態にはならないようにがんばってね。

◆X68000にWizardryを出せっていうイラストが出ていましたが，僕もそのひとりです（ちなみに会員番号A13818WIZ PLAYER'S FORUM）。でも，X68000に出るっていうんだからやっぱり凄いキャラクター，グラフィック，ダンジョン・マスターの上をいく壁，リングマスターの上をいく効果音……。なんかつまらないものになりそう。P.S.　危険物受かったぞー。

松本　勝正(17)兵庫県

◀佐藤　一　秋田県
今月のハガキの中でもっとも多かったメルヘンメイズ。個性的な絵柄でバランスよくまとまっているね。

◀橋本　茂子　東京都
もう一枚，メルヘンメイズのハガキ。すまし顔のアリスが可愛いですね。下のほうで踊り念仏(?)を踊っているうさぎがいいなあ。

画面，ゲーム性がシンプルだから，プレイヤーが想像力をふんだんに働かすことができるのがWizardryのいいところですね。

◆5月号の付録を楽しみにしています。正月のときにあった，SX-WINDOWの資料は，とても役に立っています。いままで，「dc.w $d……」などと書いていた箇所がCで組めるほど整ったわけで，今度はアプリケーションの充実を期待しています。　田中　和明(22)福岡県

そんな，他力本願なことをいわずに，積極的に参加しましょう。

◆とーとつにアーケードゲームの話でもうしわけないが，ストリートファイターIIは凄い。いま，のりにのっているカプコン。ちょっとねたんでみたいけど，やっぱり凄いものは凄い。で，なんとそのストリートファイターIIでは地面がラインスクロールしているではないか。う～ん，技術を小出しにして目立たないように地面の質感(かな？)を出すなんて……。某ゲームの「背景がゆれるから」がとってもさみしく思えた。　菅田　朋樹(17)富山県

操作に6ボタンを使用するのがネックとなりそうですが，なんとかX68000に移植できないものでしょうかね。

◆私がX68000（初代）を購入して以来，もっとも活用しているソフトは本体に同梱されていた日本語ワープロソフトです。約3年間，X68000で日記を書き続けました。今度，発売される“Multiword（マルチワード）”は本体同梱ワープロとのデータ互換性があるようなので期待しています。　佐藤　操(33)東京都

ユーザーあってのメーカーですから，きっとシャープは皆さんの期待に応えてくれることでしょう。

◆Multiwordがやっと出る。なるべくならミリ改行ができるようになってほしいな（インチよりミリのほうがよい）。CARD PRO-68Kのver2も，Musicstudio PRO-68Kも，CM-64も，ハードディスクかMOディスクもほしいし，RAMもほしい。ついでに，A列車で行こうIIIもファランクスもほしい。それにしても，X68000って周辺機器がほしくなるコンピュータだと思いませんか？　荒井　俊矢(16)長野県

で，そのうち周辺機器のなかで，いちばんほしくなるのが安い拡張スロットですよね。

◆近頃，TV放送のカラー再現性に失望を感じています。テレビ画面の外光の映り込みを減らす工夫をしたところ，コントラストが上がったのはいいのですが，放送信号のばらつきが目立つようになってしまった。各局によって違うだけならまだしも，カメラが切り替わるたびに色が違うのがわかってしまう。ひどい場合には，ひとつのCMの中で色が違ってしまうことも。チャンネルを変えるたび，番組が変わるたびに色相，色濃度を調整しなければならないなんて。いままで，私の見ていた放送はいったいなんだったんだろう。　大杉　玲(23)静岡県

ごくろうさまです。

◆Musicdrv.x & MusicI.FNCは最高だ！　デモプログラムを改造してU-220とTR-626（ドラムマシン）で鳴らしているのだが，いいですね。BASICでこんなものができるなんて。いつも，「X68000のテーマ」を流しながら生活しています。音楽をやっている人にひとこと。ドラムなどの譜割りの細かいパートは(MIDIでは)，トラックNo.の小さいほうに持ってくるとか，プログラムの頭に持ってきたほうがテンポずれが少なくなります。　畑中　道人(23)北海道

以上，畑中君のワンポイントアドバイスでした。

◆信頼できる情報筋？　によると，X68000の新モデルは16MHz版が出るそうである。そうなると，システムは変わるのか！　ソフトの互換性は？　5月号は，「ちゃだワ」+「ディスク」+「X68000の新製品？」と，なると読むのに2カ月かかりそうである。いまから大変楽しみです。　福知　健(19)京都府

5月号を読むのに夢中になって，6月号を買ってなかったりして。福知君，見てる？

◆このあいだ，川崎のアゼリアを上を見ながら歩いていたら，前の植木に頭から突っ込んでしまった。あとで一緒にいた友人に聞いたら，横のお店の店員さんが飛び出してきたり，お客が笑っていたりしたそうだ。なんともおマヌケさんな話であった。　小西山　友司(17)神奈川県

そうか，(U)氏の顔にあったあざは君のせいだったんだな。

◆この前，EDを使って作業をしていたときに，変なことが起こった。EDを起動する前にOPMファイルを鳴らしていたのだが，EDを使っている途中で曲が止まった。はずなのに，音が聞こえてくる。はじめは単音で，そして次第に音が増え，最終的にはひとつの曲として（まったく知らない曲）聞こえたのです。私は，怖くなってX68000のスイッチを切った。作業中のデータをセーブもせずに……。まさか霊……いや，バグだと信じています。　山口　大賀(17)愛媛県

そんな発想をするとは，まさか，たたられるようなことをしたんじゃないでしょうね。

◆先日，友人がシムシティーを手に入れて，ゲームを始めてしばらくしてから「どうゲームを進めていいかわからん！」といっていた。彼は最近のゲームにありがちな「展開の画一化」に慣れきっていたため，シムシティーの漠然とした設定に対処できなかったのだ。光栄のゲームなどは，シミュレーションゲームのくせにゲームの展開が，どのプレイヤーにおいても似ている。これでは，日本のゲーマーは筋書きのあるゲームしかできなくなってしまうではないか！　古川　貴(19)愛知県

そうだ！　全国○○○万人のゲームユーザーよ立ち上がれ。共に戦おう！　で，もちろんリーダーは古川君ね。

◆GAME OF THE YEARの音楽賞で，グラナダが3位というのはわかる気がするがモトスが2位，そしてラグーンが1位というのは納得できない。モトスはMIDI対応だが，曲の音楽性は低いし，ラグーンも中程度の出来である。僕の考える音楽賞1位は，誰がなんといおうと「ナイトアームズ」である。このゲーム，処理は重いが音楽性はずば抜けている。ラグーンの比ではない。おそらく，その曲を聞いたことのある人が少なかったのだろう。残念。　梅津　信幸(19)京都府

世間にアルシスの名を知らしめるため，地下組織でも作りますか？

◆この春，X68000専用の大作ADVGが2つも発売されますが，西川氏が「うっきー」と，いわない出来であることを期待します。まあ，どちらとも相当，力を入れているみたいですので，大丈夫でしょう。しかし，力を入れているせいか，どちらも高いですね。最近は，シミュレーションとアドベンチャーは1万円以上が当たり前，といった感じです。大作志向もいいですが，もっと安くて気軽に楽しめるものがあってもいいと思います。　赤城　豊和(23)兵庫県

シナリオがよければ，テキストだけでも面白いものは面白いでしょうしね。

◆春休みに入ってから本屋のバイトを始めた。しかし，これが肉体労働なのだ。どんなことをやっているのかというと，市内のいろいろな学校の教科書をひとり分ずつ科目を組み合わせてダンボールに箱詰めし，それを運んだりしている。そして，この作業をしているところが倉庫の中で，すごく寒くてほこりっぽいのだ。でも，女子高に販売に行ったりすることもある。この

◀溝畑　知幸　兵庫県
なんだかんだで9周年を迎え，お祝いのハガキもちらほら。研修中にわざわざハガキを書いてくれてありがとう。

◀岩瀬　喜代美　福岡県
ドラゴンスピリットのお兄さんをイメージしたとのことですが，耳の感じがマーベルランドのパコに似ているね。

バイトをやって，汚かった教科書や，ワゴムが2本かかっていた謎がわかった。

田中　信一(19)北海道

???，わからん。

◆僕はいま，栃木県足利市にある東足利自動車教習所というところにきているのですが，なんと，車に乗るための配車券の発行の端末に，X1 turboが使われていました。まさかこんなところで活躍しているとは思ってもいませんでした。

佐々木　信也(20)東京都

ふだん見慣れているものでも場所が違うと未知との遭遇なんですよね。

◆特集のアドベンチャーゲームの記事を読んで，「デゼニランド」「サラダの国のトマト姫」をふと思い出した。マシンはMSXからX68000へ，メディアはテープからフロッピーへ……。時の流れは早い。友達がハマッてしまって，四苦八苦しているのを救ってやったときの自分に対する賞賛の眼差しは今も忘れない。しかし，もう遠い思い出である。

高村　寿勇(17)大阪府

僕が中学生の頃は，よく近所の電器屋に友達を3人ぐらい引き連れてFM-7の「デゼニランド」を遊んだものです（和英辞典をかかえながら）。

◆最近，必要にせまられてROMライタを製作しました。前からデジタル回路の本は読んでいたのですが，初めてのハード製作でしたので，とても不安でした。なんとか完成させて，次はプログラムを組んでいます。もしも，失敗していればハードは難しいという先入観ができてしまい，二度と作ろうとは思わないようになっていたかもしれない，と考えるとゾッとします。気をよくした私は，穴を開けられROMソケットを付けられたX1GにRS-232Cを作って付けました。ケーブルも自作し，X68000と通信してます。定価の4分の1で，さらに作る楽しみも味わえるハードの製作を不用になったパソコンを使ってどんどんやりましょう。

長網　周作(22)岡山県

いまだに「ハードは難しい」という先入観を持っている人間が，ここにいたりする。

◆このあいだ自転車が原因不明のパンクになった。次の日，自転車屋さんが修理で引き取りに来たときに「きれいにしているね」と，いっていた。ことあるたびに自転車を磨いていた私は，なんだか嬉しくなった。公立高校に受かったので，いよいよX68000を買ってもらえます。Oh!Xもこれから購読していきたいと思っています。

天達　雄一(15)京都府

その調子で，X68000もかわいがってあげましょう。

◆つ，ついにX68000を買ったぞ。2年前に初めて見てから，ほしくてほしくてたまらなかったんだい。これでやっと大手を振ってOh!Xが読めるぜい。パソコンはまったくのビギナーでありますが，いままでやってきた汎用機のプログラム経験を生かして，まあ，早いとこオタッキーへの道を極めたいものです。

湊　未成美(26)神奈川県

これまた，頼もしいハガキですね。

◆やっと，4月5日にオーガスタが出る。スーパーファミコン版より遅れるのではないか，と思っていたので安心した。それにしてもリバーヒルソフトは，もうX68000にアドベンチャーを出してくれないのだろうか。黄金の羅針盤でも出してくれないかな。

浅沼　博明(21)北海道

5月号の新作情報を見て安心したかな？

◆私の愛機はX68000SUPER-HDでハードディスクは80Mバイトもありますが，購入してから6カ月たっても20Mバイトぐらいしか埋まっていません。はやく，MOドライブが必要になる身分になりたいものです。

佐藤　敏幸(25)神奈川県

それだけ，資源を有効に使っているということなのでは？

◆X68000のディスクドライブの寿命は何年ぐらいなのでしょうか。うちのは3年で2ドライブとも壊れてしまいました。修理代はドライブユニット交換で18,000円だったけど，最新式のディスクドライブはアクセス音が静かですね。

酒元　一幸(18)石川県

みんなのX68000は大丈夫？

◆いま，ちょっとしたプログラムを組んでいるが，つくづく「2Mバイト，ハードディスクなし」という環境がツライと感じられる。また，コンパイラに求められるのは「信頼性」だということも（コンパイル中にRAMディスクを破壊するな！）。ハードディスクがほしいよお。

井戸　直樹(20)岐阜県

こうして，人間はどんどん贅沢になっていくのであった。

◆このあいだ，必要にせまられてX-BASICの外部関数を作った。最初はC言語で作ろうとしたが，ポインタがわからず，BASICで作りXBAStoCで作ろうとしたが，それでもだめだった。そこで，アセンブラで作ろうと思い，村田氏の記事を読みアセンブラを少し理解してから作成を始めたら，これが動いた。いまさらになって，X68000の使いやすさがわかった。

鎗田　威(20)神奈川県

ガシガシのバリバリといわれるアセンブラですが，68000は非常にスマートなので，かえってわかりやすいかも。

◆3月18日(Oh!Xの発売日)からアルバイトを始めました。AM8：50朝礼からPM5：30終礼まで，結構，忙しいのです。仕事は再生。つまり，汚くなったパソコンなどをきれいにして，また出荷するというものです。これをやっていて，しみじみ思うことは，キーボードの掃除はこまめにやりましょう！　会社のものだからといって，粗末にしてはいけません，ということです。

折田　貴弘(18)東京都

しかし，キーボードの分解掃除はやめておこう。

◆R-360を見たとき「ついに出たか」と，思ったのは私だけではないでしょう。ところで，これからの「体感ゲーム」といわれるものは，「体感」とつくぐらいだから自分の操作するキャラクターのダメージが自分に痛みとして伝わるなんてことになったりして……（まるでプラレス三四郎みたいだなあ）。こうなれば大変でしょうね。もっとも，こんなゲームができるわけないと思いますが，10年後にはどんなマシンが登場していることやら。

藤原　彰人(20)岡山県

ゲーム世界に完全にのめり込めるようなマシンが出たら，あなた，やってみますか？

◆パソコン通信を始めてから半年がたち，近くにいごこちのよいネットができたこともあって，ほとんどはまりの状態です。今年になって，OFF会も開かれ高校生から社会人の方まで知り合いになることができました。編集部の方もいらしてみてください。きっとどこかに宣伝PICがUPされていると思いますから。ごまちゃんネットというところです。

小林　到(21)長野県

楽しそうでいいなあ。

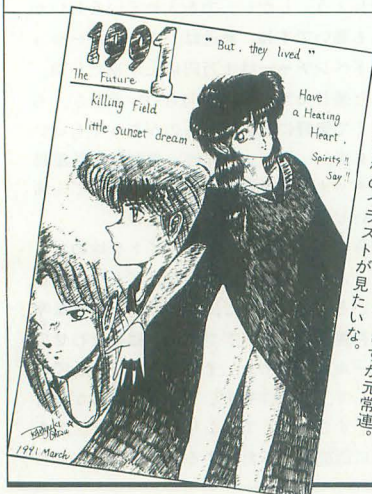
◆先日，車をミラターボTR-XXに買い換えた。もちろん中古。結構走るので，最初は喜んでいたのだがクラクション接触不良，左ドアの窓がとれるなど故障が出る出る。しかし，安さを求めて遠いところから買ったので，行くのがつらいこともあり，自分で直しました。この車のおかげで自動車修理工にでもなれそうだ。みなさん，中古車はディーラーで買いましょう。

野田　博(20)群馬県

走行中，分解したらえらいこっちゃ。

◆1年振りに部屋を掃除したら，超古代遺跡が多数，発見された。常々ほしいと思っていた本

◀大津　和之　福岡県
ファンタスティックで，おちついた感じがよく出ています。ていねいな仕上がりは，さすが元常連。今度は元気一杯のイラストが見たいな。

◀小井田　信雄　東京都
こちらは元岩手の元気少年。そうか，それは残念だったね。息抜きにイラストでも描きながらがんばってね。

（買っていたことを忘れていた）や，なくしたと思い込んでいたペンなど。やっぱり，年に2回ぐらいは掃除すべきかな……。

北野　明(25)大阪府

月に一度ぐらいは掃除しないと効果はないんじゃないかな。

◆X68000XVI発表……。自機が最新鋭機から外されてから，改めてX68000ACEに愛着がわきました。「ろくに活用できないうちにX68000ACEが老兵になってしまうのはたえられない！」てなわけで，お年玉が「パロディウスだ！」の代わりに増設RAMに化けました。最近では，X-BASICにも慣れ，コンパイラを使ってみたくなりました。「目指せプログラマ」とかいいつつ，プレゼントの希望欄にちゃっかりゲームを書いてしまう自分がほほえましい。加藤　伸一(18)神奈川県

ただゲームを遊ぶのではなく，どこが面白いとか考えながら遊んで，自分でプログラムを作るときの参考にすればいいんじゃない。

◆「大人のためのX68000」は荻窪圭氏が，スキーのため休みになったそうだが，実は私も毎週スキーに行っていたせいで，先月号のOh!Xをまだ全部読んでいません。そこで提案なのですが，冬の間はOh!Xのページ数を半分にするというのはどうでしょうか。そうすれば，Oh!Xのスタッフも読者もこころおきなくスキーに行けるってもんです。　横田　紀明(24)山口県

なるほど，そうなると春は花見シーズンで半分，夏になれば夏休みで半分，秋は食べ歩きのため半分にする。いいなあ（でも給料が半分になったらしゃれになんない）。

◆4月某日，会社で日経新聞をボーッと眺めていた私の目は，点になってしまった。X68000XVIとはいったいなんなんだあ。「CPUに16MHzの高速モードを搭載」と，書いてあったが……。

原田　真志(20)静岡県

◆X68000の新型が出たぜー。ほしいぜー。しまった，俺は今年も浪人だ！

竹沢　裕利(19)千葉県

◆1.6倍だと！　むむむ……。まあ，32ビット化じゃないからいいか（ハンパな32ビットはいらないしね）。　吉村　昇(19)大阪府

◆ああ……。とうとう恐れていたことが。X68000XVIの記事を見て，世代交代の時期になったのを実感しました。ところで10MHzと16MHzの違いはどうなんですか。アイルトン・セナと中嶋悟ぐらいの違いかな？　日本がんばれ！

▲金子　聡　新潟県
これまた，かわいらしいダンジョン・マスターのイラスト。ダンジョンがこんなイメージだったら冒険も楽しくなりそう。

佐藤　泰堂(16)東京都

4月号のハガキの中からX68000XVIについて書いてあるものをいくつか拾ってみました。ほかにも，まったく知らなかった人，どこからともなく情報を入手した人，見当違いの憶測をしている人などさまざまでしたが5月号の新製品紹介を見てみんなはどう思ったかな。

ぼくらの掲示板

●掲載ご希望の方は，官製ハガキに項目(売る・買う・氏名・年齢・連絡方法……)を明記してお申し込みください。
●ソフトの売買，交換については，いっさい掲載できません。
●取り引きについては当編集部では責任を負いかねます。
●応募者多数の場合，掲載できない場合もあります。
●紹介を希望されるサークルは必ず会誌の見本を送ってください。

仲間

★我が東開発塾では，このたびユーザーの皆様方とのコミュニケーションを取るにあたって，会報を作ることになり，会員を大募集しています。主な活動は会員の皆様が作り上げていく月1回のディスク会報です。初心者から上級者まで幅広く募集しますのでお気軽にどうぞ。詳しくは62円切手と返信用封筒を同封のうえ，下記の住所まで送ってください。〒870-02　大分県大分市大字城原字千町2368-2第41久永コーポ313号　濱田　憲明(18)

★X68000ユーザーを対象とした「サークル・IN」では会員を募集しています。活動は1月半に1回のディスクマガジンの発行です。また，PDSや配布可能なフリーウェアをつめたディスクを，希望者に無料で配布するサービスも行っています。入会金は無料。会費はディスクマガジン発行時の300円のみです。なお，一緒に活動しようという気のない方の入会はご遠慮願います。活動する気があるのなら初心者の方でも大歓迎です。入会したいという方は，まず会報の見本(最新号）を無料でお送りいたしますのでフォーマット済みのディスク＋返信用封筒＋120円切手を同封のうえ，下記の住所までお送りください。〒440　愛知県豊橋市西小鷹野2-10-1　高辻力也

売ります

★熱転写カラー漢字プリンタ「CZ-8PC3」を送料込み2万～3万円で。箱，マニュアルあり。連絡は往復ハガキでお願いします。〒730　広島県広島市中区大手町3-12-19　西口　秀幸

★1Mバイト増設RAM，「CZ-6BE1A」（X68000ACE以降用）がまぬけな理由により余分が出てしまったのであげます(送料はそちらもち)。ただし動作チェックはしていません。改造して使うか，そのまま使うかを明記して，往復ハガキで連絡してください。なるべく貧乏な人を優先したいと思います。〒470-01　愛知県愛知郡東郷町春木白土1-768　山口　青星

買います

★カワイの音源モジュール「PHm」を送料込み3万円以下で買います。完動品であればキズ，汚れは問いません。だだしMIDIケーブルとマニュアルだけは必ず同梱してください。また，エフェクタ（ディレイか，リバーブ）をつけてくださる方には7千円，電波新聞社の「オーディオミキシングケーブル」をつけてくださる方には千円プラスします。安い人優先。連絡は往復ハガキでお願いします。〒384-23　長野県北佐久郡立科町大字牛鹿1230　小宮山　博志(17)

★X68000PRO用のI/Oデータ機器製か，シャープ純正の増設1Mバイト RAMボードを1万5千円以内で。また，X68000対応のMIDIボード＋ローランド製のMT-32以降の音源を6万～7万円あたりで。できればマニュアル，箱つき（キズあり可)。連絡は往復ハガキのみで。〒306-06　茨城県岩井市神田山1377-1　倉持　哲也(16)

★X1用カラーイメージボードII「CZ-8BV2」と立体映像セットを各1万円以下で。完動品，付属品付きで送料は当方で負担。連絡はハガキで。〒330　埼玉県大宮市深作1856-3東五番街2-401　馬場　俊行(34)

★アスキーの「X68000テクニカルデータブック」か，POPCOMの「X68000データブック」を定価で売ってください。連絡は往復ハガキで。〒260　千葉県千葉市磯辺3-12-10山川　秀幸(21)

バックナンバー

★Oh!Xの1989年1月号，3月号，4月号を各1,500円でお譲りください（切り抜き不可)。連絡はハガキでお願いします。〒813　福岡県福岡市東区唐原1-4-12　森　浩(29)

編集室から from E・D・I・T・O・R

DRIVE ON

このコーナーでは，本誌年間モニタの方々のご意見を紹介しています。今月は4月号の内容に関するレポートです。

●「決定！　1990年度GAME OF THE YEAR」はページ数が少なくなんとなく盛り上がりに欠けた感じがする。「勝手にGAME OF THE YEAR」が少なかったのも残念。1つひとつの賞を重視という主旨だったようで，それも重要だとは思いますが，ボリューム不足もつまらない。このへんのバランスがむずかしいかもしれないけど，来年もガンバッテください。
畑　剛志(19)　X1turboZII，MSX/2，JR-100　北海道

●「アマチュアCGAコンテスト入選作品発表」。写真の点数が少ないという問題については，本来CGAは動画であり，動いているところに価値があると思うので，静止画面の写真を並べてもあまり意味がなく，むしろ今回のようにポイントとなる場面にしぼって写真を紹介するほうがいいと思います。
泉　昭彦(21)　X1turbo,PC-E500　東京都

●4月号の特集ではただのゲーム特集ではない，なみなみならぬものを感じてしまいました。「あれがいい，これがよくない」といった単なる紹介に終わらず，ここまで考えることが重要なんだということに気がつきました。これからもこのような「こだわり」を持ちつづけてほしいものです。

また，「吾輩はX68000である」はとっても楽しかったですよん。実は私，Cを持て余し気味だったんですが，これを読んでいてアセンブラに手を出し，村田氏のあの本のChapter 03_Hまでいっています（三日坊主にならないなんてめずらしい)。初心者向けに，むずかしくなく，楽しく，X68000の魅力を見せていく連載になればと思っています（なんとまあ，お高い望みだこと)。でも，お願い！　Cだけはやめて。あえていうならBASICに走るのもナシにしてくださいね。そういう主旨のものではなさそうだから，そのへんは大丈夫でしょうけど。
安井　百合江(16)　X68000 PRO　愛知県

●新連載の「よいこのSX-WINDOW講座」は難しいことをやっているのですが，それを難しく見せないところには脱帽です。とりあえず，ここに掲載されているプログラムを打ち込めば，SX-WINDOWで制御ボタンを出すことができるようになるのですね。これは，よく考えるとすごいことです。にもかかわらず，すらすら〜と読めてしまいます。コンパイルの方法や環境整備，バッチファイルの紹介，フローチャート，バグ出しなど，苦労して手に入れたものを無駄にしないよう，本気で「理解してもらおう」となさっている様子が，中森さん自身の熱意とともに伝わってくるようです。楽しい連載が始まりました。不明な点は克服して，我々の道標になっていただきたいと思います。がんばってください。
浅野　憲(19)　X68000 PRO，X1turbo III，X1F，MZ-80C，FM-77L2，M5Jr.，PC-6001，PC-1245　大阪府

●「シミュレーションプログラミング入門」のおかげで，ファジィ理論がどんな代物なのかやっとわかりました。こういうその筋な人の話というのは非常にためになるので，バンバンやっていただきたいなと思います。でも私個人としては，PID制御くらいならいざしらず，ファジィ制御になってくると，モデリングも手に負えない，そんな気がして手も出せないというのが正直な気持ちです。
高村　信(20)　X1turbo,PC-8801mkII　東京都

ごめんなさいのコーナー

5月号　特別付録　KORG M1用音色データ

音色設定がうまくできませんでした。リスト1のプログラムを実行して，各データをデバッグしてください。

5月号　実数型コンパイラ言語REAL

OFFSET命令が正常に動作しないようです。下記のように修正してください。

```
3918 CD 78 69

39D2 CD 81 69

504B CD 81 69

5B7D FD E5 D1 2A C5 33 19 EB : D9
5B85 2A CC 33 7B 95 7A 9C 38 : 87
5B8D 15 7C C6 10 67 7B 95 7A : 58
5B95 9C DC 43 61 2A 6A 1F 7B : 4A
5B9D 95 7A 9C D4 43 61 08 12 : 3D
5BA5 FD 23 CD AD 4F          : E9
---------------------------------
SUM: 6A A6 76 97 7D F3 71 2A 174E

6978 ED 5B C5 33 19 0B 5E 0A : CC
6980 C9 ED 5B C5 33 19 5E 71 : F1
6988 23 C9                   : EC
---------------------------------
SUM: D9 11 20 F8 4C 24 BC 7B 99D8
```

リスト1

```
10 /* M1 data debug
20 int fn1,fn2
30 char data(16000)
40 str fname
50   input "filename:";fname
60   f1=fopen(fname+".m1","r")
70   f2=fopen(fname+".m2","c")
80     fread(data,fsize(f1),f1)
90     data(&H37F7-2)=0
100    fputc(&H92,f2):fputc(&H0,f2)
110    fwrite(data,fsize(f1),f2)
120  fcloseall()
130 /*frename(fname+".m1",fname+".mo")
140 /*frename(fname+".m2",fname+".m1")
150 end
160 func fsize(fn)
170 int a,b
180   b=fseek(fn,0,1)
190   a=fseek(fn,0,2)
200   fseek(fn,b,0)
210   return(a)
220 endfunc
```

バグに関するお問い合わせは
☎03(5488)1311(直通)
月〜金曜日16：00〜18：00

お問い合わせは原則として，本誌のバグ情報のみに限らせていただきます。入力法，操作法などはマニュアルをよくお読みください。
また，よくアドベンチャーゲームの解答を求めるお電話をいただきますが，本誌ではいっさいお答えできません。ご了承ください。

年間モニタ決定
求む，技術スタッフ
よろしく

▶今月号は創刊9周年記念号ということで，特別企画満載，増ページ，特別定価600円でお送りしました。また，6周年を迎えたS-OSには，前々から予告していた待望の「Small-C処理系の移植」が掲載されています。

▶Oh!Xでは技術協力スタッフを募集しています。住所・氏名・年齢・電話番号を明記のうえ，Oh!X編集部「スタッフ希望」係までご連絡ください。特に，アセンブラ，C言語を使える方を望みます。

▶昨年度は愛読者年間モニタの応募者数が少なくて苦労したのですが，今回は本当に多くの方が応募してくれました。

相沢　道造（東京都），朝野　貴敦（滋賀県），市川　徳明（東京都），伊藤　政弘（愛知県），遠藤　隆一（北海道），岡本　洋明（静岡県），奥山　貴士（奈良県），金子　聡（新潟県），櫛田　宏紀（大阪府立大手前高校理化学研究部代表），国政　寛（大阪府），功刀　和久（埼玉県），小林　紀孝（東京都），佐藤　哲也（北海道），宍戸　輝光（東京都），高津　陽一（兵庫県），高辻　力也（愛知県），高橋　毅（埼玉県），谷口　有香（北海道），坪井　秀次（静岡県），弦元　達也（香川県），内藤　陽一（愛知県），中村　圭介（神奈川県），中村健（埼玉県），野原　志貴乃（埼玉県），平木敬太郎（福井県），藤本　冬彦（神奈川県），前田　秀樹（京都府），増山　修（長崎県），松本　知己（石川県），松本　康裕（広島県），水沼　一英（群馬県），安岡　毅（京都府），山岡　伸一（大阪府），山崎　一茂（岡山県），山森　和博（愛知県），安井　百合江（愛知県）

以上36名の皆さんにモニタを務めていただきます。来月号からレポートをお送りしますので，よろしくお願いします。昨年度の年間モニタの方々は今月号で最後のレポートとなります。ありがとうございました。

▶今月の「猫とコンピュータ」，「シミュレーションプログラミング入門」はともに筆者多忙のためお休みさせていただきます。

投稿応募要領

●原稿には，住所・氏名・年齢・職業・連絡先電話番号・機種・使用言語・必要な周辺機器・マイコン歴を明記してください。

●プログラムを投稿される方は，詳しい内容の説明，利用法，できればフローチャート，変数表，メモリマップ（マシン語の場合）に，参考文献を明記し，プログラムをセーブしたテープ（ディスケット）を添えてお送りください。また，掲載にあたっては，編集上の都合により加筆修正させていただくことがありますのでご了承ください。

●ハードの製作などを投稿される方は，詳しい内容の説明のほかに回路図，部品表，できれば実体配線図も添えてください。編集室で検討のうえ，製作したハードが必要な場合はご連絡いたします。

●投稿者のモラルとして，他誌との二重投稿，他機種用プログラムを単に移植したものは固くお断りいたします。

あて先

〒108 東京都港区高輪2-19-13 NS高輪ビル
ソフトバンク出版部
Oh!X「㋢㊀㋮㊔」係

S H I F T · B R E A K

▶昨日，浦和市は原山団地を自転車で流していたところ，団地の窓に見覚えのあるマシンが？　おお，そこにはX1Fを目の前にもくもくとプログラムを打ち込むお子様の姿がはっきりと。何度もRUNしては首をかしげ一生懸命悩んでる様子でした。思わず自分にもそーゆー時代があったなあと自転車に乗ったまま心の中で応援してしまいましたとさ。（哲）

▶ショートプロの原稿を書いた2日後。都合で神奈川の下宿を引き払い，3年ぶりの実家に帰ってきました。さらばひとり暮らし。これでおちついて原稿が書けるな，と思っていたら早速こっちの友達がやってきて某“いただきストリート”なる多人数ゲームをやっている。これは友達の選び方とかなんとかというよりは……運命なのかもしれない。（で）

▶対戦ボンバーマンでは「青いバカ」と罵られ，対戦ポピュラスで連戦連敗だった私だが，まだまだ世の中捨てたものではなかった。ああ，私にはオーガスタがあった。先日の浦川君，山田君，西川の善ちゃんなど，強者どもと一緒に戦ったトーナメントでは，涙々の逆転勝利！　はっきりいって「ざまあみろ！」だ。（チップイン146ヤードの毛）

▶ちょいとわけありで，計算機をしまいこむはめに陥った。遠くへ行くので3カ月は愛機に触れない。社会復帰を果たしたときには浦島太郎になってやしないかと心配だ。キーボードと泣き別れになった指は早くもさびしがっている。僕がX68000に携帯機種もほしいと切実に思ったのは今回が最初。これが最後になることを願いたいものだ。（A.T.）

▶こんな名前絶対に覚えられないと思ってたのに，いつのまにか覚えてるんだもんな。ベススメルトヌイフ，ベススメルトヌイフ，ベススメルトヌイフ。すらすらいえる。カタカナ9文字だぜ。メモリの無駄だって。できるものなら「メリットはジンクピリチオン配合」とか「クレアラシルはサルファレゾルシン処方」とかいうのと一緒に忘れたいよ。（Mu）

▶要するに，記号であるという発見が新鮮であったがゆえに記号で遊ぶ行為により深い感情をこめられたわけだが，前段階を経ずして遊ぶことだけが先行している今の状況は，使い回してひっくり返すことしかできない貧しさを表し，まだ開いてみれば別のものが見えるだろうに，眺める角度は変えても深さは変えようとしない気の小ささである。平和。（K）

▶4月19日，待ちに待った「パロディウスだ！」の発売日。もしかしたら売り切れかも？　という焦りがその日の膨大な残業を放棄させた。渋谷の某店に山のように積まれたパッケージを見てそれが杞憂だったとわかったが，買い損ねていたら一生後悔しただろう。ゲームの出来はGOODで，上司からの非難と引き替えにするだけの価値は十分あった。（KO）

▶徹ボン，徹オーガスタ，こういう余計なことをしていると必然的に家に帰るのが遅くなる。そんな時間の横浜駅はとても面白い。酔っ払いに常識はない，の言葉どおり路上で座り込みカンビールを開けて宴会したり，路上のケンカ，まあこれくらいは普通。このあいだは，なにを思ったのか腕立てふせをしているヤツを見かけた。体力余ってんのかな。（J）

▶先日，ハンブル・パイの「パフォーマンス」を聞きながら新聞を読んでいると，メンバーだったスティーブ・マリオットが焼死したと死亡欄に小さく載っていた。単なる偶然だが，このようなささいな偶然によって物事のインパクトはかなり左右されるのだなあ，とあらためて思いつつ，故人の絶頂期にあたるこのアルバムに聞き入ってしまった。（A）

▶新入社員で（J）が入ってきた。「ああ，やっとこれで5人だ。少しは仕事が減るかな？」などと淡い期待を抱いていたのだが，編集長の「人が増えたら仕事も増えてくもんだ」のひと言で，それはもろくも崩れ去った。やっぱり結婚して主夫をもらおう，と真剣に考えている今日この頃。「掃除と洗濯よろしくね」「あいよ」これって絶対いい！（E.O.）

▶編集後記をのぞき込む。「え，（純）？」「……じゃ（J）にします」。そう，Jだね。Yはザコだし。TやUよりは強力になってもらわなければ困るのである。そういえば昔は@氏は主役だったし，M氏は人を惑わし，N氏はつつくとメシをおごってくれた。うーむ。今度はやっぱりGかDのイニシャルの新人がほしいと思う。（U）

▶で，注目の（J）君だが，このコーナーでも私の隣（←ほれ）におさまったものの，ご覧のとおり学生ライターの気分が抜けていない。客「え，純くん編集部入ったの？」J「ええなんとかレベルアップしました」T「こらこらクラスチェンジでレベルは1からだろ」J「うん」A「こらこら，うんはダメやぞ，うんは」J「は～い」。やれやれ。（T）

microOdyssey

家から電車を乗りつぐこと3回，4時間半かけて実に14年ぶりに波久礼の駅に降りた。“はぐれ”と読む。まさにその名のとおり，埼玉のはずれにある小さな駅だ。いまは違うが，当時は電車のドアがひとつ（しかも手動で）しか開かなかったぐらい，人の乗り降りが少ない駅だ。

目の前に懐かしい風景が広がる。ほんのちょっとだけ道路は広くなっているほかは，14年前となにも変わらない緑の山。荒川の流れも昔のままだ。細く曲がりくねった道を10分ほど歩くと目的地に着く。埼玉県立寄居養護学校，それがそこの名称だ。病院と学校が一体となった，いわゆる虚弱児のための養護施設。あたしは小学校の高学年をこの学校で過ごした。養護学校は，木造平屋建てから，どデカい4階建ての鉄筋コンクリートへと様相を変えていた。歩くときしむ古い床や，いまにも落っこちてきそうな茶色の瓦の面影は，もうどこにもなかった。

が，ふと視線を右へ向けると，昔とまったく変わっていないものを見つけた。グラウンドと裏山だ！　山を切り崩して作られたせまくるしいグラウンドは，もうほとんど使われていないのか，草はぼうぼう，鉄棒は取り外されているといった惨状だった。が，かつて引いていたローラーはいまでもドデンと居坐っていた。そこに昔の自分を見ることができたようで，とてもうれしかった。けもの道から裏山へ登る。が，えらい急なので登りきるのに20分もかかってしまった。こんな山道を駆け回っていたのかと思うと，ここにいた頃のほうがよっぽど健康だったんじゃないか，と思わず苦笑してしまった。

なんだかすべてが懐かしくてうれしかった。

最初ここに放り込まれたときは，あたしは養護学校の名を借りた鑑別所かと思った。ケンカやリンチは日常茶飯事，看護婦は見て見ぬふり，もちろん常識なんてものは一切通用しなかった。かくいうあたしも入って3カ月間，東京モンというだけでメシは残飯，水はぶっかけられるわ殴られるわと，さんざんな目にあった。いまだに右足のふくらはぎにタバコの跡があるくらいだ。でも，いまとなっては単なる懐かしい思い出だ。人格形成期にここにいたことは，非常にプラスだった。絶対生き抜いてやる，そういう強い意志を持つ機会があったのだから。いまでも“あそこであんなに頑張れたんだから”と思って踏ん張ることが多い。ゆえにあたしにとっての故郷は，いつでもこの養護学校だ。

とはいえ，いままではこの場所になるべくなら来たくないと思っていた。常に突っ走ることで自分を維持してきたあたしは，弱かった頃の自分に会いたくなかったから。でも，そういう自分の弱さを認めて，振り返ることも大事だと教えてくれた人がいる。大切にしてくれているんだと思う。そろそろターニング・ポイントなのかもしれない。わからなくなったら，最初の純粋なところに帰ってみよう，いまやっとそう思えるようになった。

さてと，裏山の向こうに第2の目的地，少林寺がある。なぜ寺が目的かというと，養護学校時代に借りた（くすねたともいう）お金を返すためだ。お金を持つことを許されなかったあたしらは，そこでお金をちょうだいして飢えをしのんでいたのだった。ああ，住職さん，ごめんなさい。仏様は困った人の味方よね。　(E.O)

1991年7月号6月18日(火)発売

特集 Personal Tool, BASIC

MAGIC. FNC/MAGICLIB.A

tinyCalc応用編

X1用シューティングゲーム　DEFEAT2

特別付録　*X-BASIC簡易リファレンスブック*

特別定価　600円

バックナンバー常備店

東京	神保町	三省堂神田本店5F 03(3233)3312
	〃	書泉ブックマートB1 03(3294)0011
	〃	書泉グランデ5F 03(3295)0011
	秋葉原	T-ZONE 7Fブックゾーン 03(3257)2660
	八重洲	八重洲ブックセンター3F 03(3281)1811
	新宿	紀伊国屋書店本店 03(3354)0131
	高田馬場	未来堂書店 03(3200)9185
	渋谷	大盛堂書店 03(3463)0511
	池袋	リブロ池袋店 03(3981)0111
	〃	西武百貨店9F コンピュータ・フォーラム 03(3981)0111
神奈川	横浜	有隣堂横浜駅西口店 045(311)6265
	〃	有隣堂ルミネ店 045(453)0811
	藤沢	有隣堂藤沢店 0466(26)1411
神奈川	厚木	有隣堂厚木店 0462(23)4111
	平塚	文教堂四の宮店 0463(54)2880
千葉	柏	新星堂カルチェ5 0471(64)8551
	船橋	リブロ船橋店 0474(25)0111
	〃	芳林堂書店津田沼店 0474(78)3737
	千葉	多田屋千葉セントラルプラザ店 0472(24)1333
埼玉	川越	黒田書店 0492(25)3138
	川口	岩渕書店 0482(52)2190
茨城	水戸	川又書店駅前店 0292(31)0102
大阪	北区	旭屋書店本店 06(313)1191
	都島区	駸々堂京橋店 06(353)2413
京都	中京区	オーム社書店 075(221)0280
愛知	名古屋	三省堂名古屋店 052(562)0077
	〃	パソコンΣ上前津店 052(251)8334
	刈谷	三洋堂書店刈谷店 0566(24)1134
長野	飯田	平安堂飯田店 0265(24)4545
北海道	室蘭	室蘭工業大学生協 0143(44)6060

定期購読のお知らせ

Oh!Xの定期購読をご希望の方は綴じ込みの振替用紙の「申込書」欄にある「新規」「継続」のいずれかに○をつけ，必要事項を明記のうえ，郵便局で購読料をお振り込みください。その際渡される半券は領収書になっていますので，大切に保管してください。なお，すでに定期購読をご利用の方には期限終了の少し前にご通知いたします。継続希望の方は，上記と同じ要領でお申し込みください。

海外送付ご希望の方へ

本誌の海外発送代理店，日本IPS（株）にお申し込みください。なお，購読料金は郵送方法，地域によって異なりますので，下記宛必ずお問い合わせください。

日本IPS株式会社
〒101 東京都千代田区飯田橋3-11-6
☎03(3238)0700

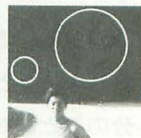
Oh!X 6月号

■1991年6月1日発行　特別定価600円（本体583円）
■発行人　孫　正義
■編集人　橋本五郎
■発売元　ソフトバンク株式会社
■出版事業部　〒108 東京都港区高輪2-19-13　NS高輪ビル
Oh!X編集部　☎03(5488)1309
出版営業部　☎03(5488)1360 FAX 03(5488)1364
広告センター　☎03(3297)0181
■印　刷　凸版印刷株式会社

満開の電子ちゃん

作：いわい いっぺい
え：岡村祭

高野正巳（神奈川県）

満開製作所発行「月刊・電脳倶楽部」…この怪しげな響きを持つ"雑誌"の存在を知ったのは、Oh！Xの広告からでした。その名前から警戒して、とりあえずは6千円だけ払い半年間の契約にしました。しかし、届いた実物をみてビックリ。ディスクの中にぎっしりつまった便利なツールや楽しいビープ音によって、私のパソコン環境は一変してしまいました。さらに掲載プログラムはほとんどソース付きなので、最近始めたアセンブラの学習にも大変役に立っています。今では、私も立派な年間購読者（中毒患者）です。

赤えんぴつならゴールが見える!!

赤えんぴつ（JRA版）

最近甘口の予想ばかりとお嘆きの貴兄に、辛口の予想をデータから導く「赤えんぴつ」をそんな貴方にお送りします。
今迄の競馬のコンピュータ用予想プログラムは、オッズを入力して予想するものばかりでした。
この方法はデータ数が少なく入力し易いのですが、オッズは馬券を買った人たちの人気投票的なものですし、貴方の個人的な御意見等も反映出来ず、堅い馬券は時々当たるものの、中穴以上になると7点ぐらい予想をしてもはずれる事が多々あり、回収率も１００％を割るものばかりでした。
今回発売した「赤えんぴつ」は当たる馬券を予想するのでは無く、予想紙に載っている馬の過去のデータを入力して、ゴールする時のタイムを予想し上位３頭の馬から３点の組み合わせをはじき出します。
当社で行った過去９０回のレースを模擬的に各レース３点で予想した結果では３５％の的中率を出し、回収率も１３０％を上回っています。
過去のデータだけを入力するのでは無く、最新の馬の調子や馬場状態等の主観的なデータも１０～１００％の数字に置き換えて予想に反映させたり、それらのデータをディスクにセーブする事が出来ますから、レースの前日にデータを入力しておき、レース当日の天候等、直前の情報で各馬のデータを修正して予想を立て直す事も出来ます。
又、コンピュータの苦手な方でも簡単にデータの入力が出来る様にカーソルコントロールキーと実行キーの５つのキーを使うだけで総ての操作が出来ます。
このプログラムはＪＲＡ主催の全国１０ケ所（札幌、函館、福島、新潟、中山、東京、中京、京都、阪神、小倉）の各競馬場以外の公営競馬場では使えません。

赤えんぴつ	X68000用　2HD	20,000円

便利な超高速通信機能付で、DB．Xよりも使い易く、X1 turbo のディスクもアクセス出来る。

SUPER DEVICE MONITOR "T"	X68000用　2HD	15,000円

X68000 と超高速通信が出来てMS－DOSのディスクや内部増設ＲＡＭにもアクセス出来る。

SUPER DEVICE MONITOR "T"	X1 turbo用　2HD／2D	13,000円

＊MS－DOSはマイクロソフト社の商標です。
＊商品の価格には消費税は含まれていません。

▶お求めは全国の有名マイコンショップでどうぞ。
通信販売をご希望の方は当社へ直接、商品名・機種名・メディア名・住所・氏名・電話番号を明記の上、現金書留にてお申し込みください。（送料無料）

BLUE SKY Co.
株式会社　BLUE SKY
〒411　静岡県三島市加茂16-4 ☎0559-72-6710

最大メモリ8Mバイト

KGB-X68PRKⅡ

新発売!

● 8M増設メモリと数値演算プロセッサが1枚のボードに収まります。
●従来品(KGB-X68PRK)に比べて大幅なコストダウン。
●メモリ容量2M/4M/6M/8Mの4種類、それぞれに数値演算プロセッサ有無のモデルを用意しました。
●数値演算プロセッサ無しモデルではMC68881RC16の購入で簡単にグレードアップが可能です。
●当然、2M/4M/6Mモデルでは購入後も8Mまでのメモリ増設が可能です。

品名	価格
2Mメモリ数値演算プロセッサ無し	¥ 55,000
4Mメモリ数値演算プロセッサ無し	¥ 90,000
6Mメモリ数値演算プロセッサ無し	¥ 125,000
8Mメモリ数値演算プロセッサ無し	¥ 160,000
2Mメモリ数値演算プロセッサ付属	¥ 85,000
4Mメモリ数値演算プロセッサ付属	¥ 120,000
6Mメモリ数値演算プロセッサ付属	¥ 155,000
8Mメモリ数値演算プロセッサ付属	¥ 190,000

PRKII質問箱

Q 購入後のメモリ増設はどうやるのでしょう?
A ご購入後のPRKⅡに対するメモリも増設は半田付けなどの技術を要するためボードを当社に送り返していただき増設をいたします。ご自分でメモリ増設をする場合には部品の販売も予定しております。

Q 数値演算プロセッサにMC68882を使用することは可能ですか?
A MC68882では動作しないソフトが存在するために使用することは出来ません。

Q 旧PRKとPRKⅡではどこが違うのですか?
A 1枚に収まるメモリが最大で8Mになった以外は同じです。

Q 数値演算プロセッサを使うと速度が速くなるのですか?
A 数値演算プロセッサを使用することにより速くなるのは実数演算のみです。画面表示などは速くなりません。

充実のBASIC HOUSEソフトウェア&ハードウェア

品名	価格
高速12BIT, 16CH A/Dコンバータボード(KGB-AD12) X1	¥118,000
フォトアイソレーション16BITデジタル入出力ボード(KGB-PIO) X1	¥ 42,000
アイソレーション16BITデジタル入出力ボード(KGB-X68PIO) X68000	¥ 68,000
ハンディプリンタ&インターフェース(HANDYPRINTjack) X68000	¥ 24,800
高速12BIT, 4CH D/Aコンバータボード(KGB-DA4) X1	¥ 98,000
汎用ローコストA/D&PIOボード(KGB-X1S) X1	¥ 19,800
高速12BIT, 16CH A/Dコンバータ(KGB-X68ADC) X68000	¥128,000
64180CPUボードMach 180(KGB-CPXB) X68000	¥ 98,000
ローコストMIDIインターフェース(MELODY BOX) X68000	¥ 16,800

BASIC拡張関数パッケージ(B6-6301) ¥9,800　C言語ライブラリ(B6-6305) ¥6,800
ディスクキャッシャー(B6-6304) ¥6,800　Toys & Tools (B6-6307) ¥6,800
BASIC拡張関数パッケージC言語ライブラリ付(B6-6306) ¥14,800
アイコンエディタ(B6-6303) ¥4,800　CP/M68Kエミュレータ(B6-6302) ¥19,800

おしらせ

スタッフ募集!

計測技研/First Class Technologyでは、プログラマースタッフを募集しています。
X68000大好き人間、新しい物好きの明るい人、いっしょに開発しましょう。

ご希望の方は、計測技研高橋までご連絡下さい。

ビデオボードを外付けに!!
ビデオボードケース(KGB-BVBX)
大好評発売中 定価9,800円

SHARPより発売されているCZ-6BV1を外付けにするケースです。このケースの使用によりあなたのX68000のスロットが開放されます。

Human68k下のソフトのCRT出力を強制的に15kHZ出力にする(768×512モード除く)おまけユーティリティ付き

全国どこでも発送可　長期クレジットOK　送料全国均一¥1,000　宅配便にて即日配送

株式会社計測技研
本社営業部/マイコンショップ/通販部　宇都宮市竹林町503-1　TEL0286-22-9811　FAX0286-25-3970
大田原営業所/マイコンショップ　大田原市美原1-13-4　TEL0287-23-5352　FAX0286-23-5364
マイコンショップ BASIC HOUSE　お申し込み・お問い合せは ☎0286-22-9811(代)

パソコン
ワープロの
ことなら
なんでも！

デ

株式会社 デンキヤ

〒332 埼玉県川口市西川口4丁目6番4号
AM11:00〜PM7:00 水・木定休

今月の超特価品

シャープ
X68000セット
SURER

特価

TEL

★X6800本体★

品番	価格
CZ-603C	¥
CZ-613C	¥
CZ-653C	¥ 192,400
CZ-663C	¥
CZ-623C-TN	¥ 336,200
CZ-604C-TN	¥ 234,900

★X6800ディスプレイ★

品番	価格
CZ-606D	¥ 53,900
CZ-613D	¥ 91,100
CZ-605D	¥ 77,600
CZ-604D	¥ 64,000
CU-21HD	¥ 99,900

★プリンタ・ケーブル付★

品番	価格
CZ-8PG1	¥ 90,400
CZ-8PG2	¥ 111,200
CZ-8PK10	¥
CZ-8PC4	¥
CZ-8PC5	¥ 67,300
IO-735X	¥
CZ-6PV1	¥
HG-4000	¥ 140,600
VP-2600	¥ 104,400
VP-960	¥ 83,800
VP-1600	¥ 87,500
VP-1350	¥ 62,400
VP-550	¥ 53,900
LP-3000	¥
LP-7000G	¥
AP-900	¥ 62,400
AP-600	¥ 47,000

★ハードディスク各種★

品番	価格
CZ-620H	¥
CZ-64H	¥ 90,000
IT X80S	¥ 92,800
IT X130S	¥ 114,600
IT X640	¥
IT X680	¥
HXD040	¥
HXD042	¥
AV-090WS	¥ 116,800
AV-050WS	¥ 93,100

★インターフェイス各種★

品番	価格
CZ-6BS1	¥ 22,400
CZ-6BM1	¥ 20,100
CZ-6BV1	¥ 15,800
CZ-6BF1	¥
CZ-6BG1	¥
CZ-6BU1	¥
CZ-6BC1	¥
CZ-6BL1	¥
CZ-6BL2	¥

★RAMボード★

品番	価格
CZ-6BE1B	¥ 21,000
CZ-6BE2	¥
CZ-6BE4	¥
P10-6BE1-A	¥ 18,100
P10-6BE2	¥ 33,800
P10-6BE4	¥ 59,400

★その他★

品番	価格
CZ-6BP1	¥
CZ-6EB1	¥

★モデム各種★

品番	価格
MD24FS5	¥
MD24FS7	¥ 45,000
MD24FP5 II	¥ 29,700
PV-M24VM5	¥ 29,700
PV-M24	¥ 27,700
コムスターズ2424/5	¥ 27,800
コムスターズ2424/4	¥
SR-120S	¥
SR-240S	¥
SR-240V	¥

★ソフト各種★

品番	価格
CZ-249GS	¥ 22,400
CZ-255GS	¥ 6,600
CZ-256GS	¥ 6,600
CZ-245LS	¥ 33,600
CZ-260LS	¥ 7,400
CZ-251BS	¥ 29,900
CZ-243BS	¥ 14,900
CZ-240BS	¥ 11,100
CZ-259SS	¥ 5,100
CZ-257CS	¥ 14,900
CZ-219SS	¥ 22,400
CZ-252MS	¥ 21.600
CZ-213MS	¥ 14,100
CZ-247MS	¥ 21,600

★ゲームソフト各種★

24時間テレホンサービス
0482-54-3444

お申し込みはお電話で

TEL 0482-54-3400
FAX 0482-54-3443

★振込先★
三菱銀行西川口支店
普通0258081
㈱デンキヤ

SHARP
パソコン本体から周辺機器まで品数取り揃え
大特価セール実施中!!

型名	品名	正価	特価
PC-E500BL	ポケコン	28,800	19,500
PC-1600K	ポケコン	69,800	49,800
PC-1360K	ポケコン	36,800	32,800
PC-1360	ポケコン	29,800	19,800
PC-1262	ポケコン	24,800	19,600
PC-1248DB	ポケコン	11,000	9,800
PC-1280	ポケコン	24,800	19,600
CE-T800	ポケコンRS-232Cコンバーター	12,800	11,800
CE-203M	ポケコンRAM32K	32,000	7,000
CE-202M	ポケコンRAM16K	35,000	6,000
CE-201M	ポケコンRAM 8K	18,000	3,000
CE-1600M	ポケコンRAM32K	32,000	16,000
CE-1600F	ポケコンフロッピードライブ	39,800	34,800
CE-1600P	ポケコンプリンター	69,800	59,800
CE-1650F	ポケコンDISK	9,800	8,800
CE-161	ポケコンRAM16K	50,000	3,800
CE-1601M	ポケコンRAM64K	45,000	30,000
CE-1600E	ポケコンディスクインターフェイス	19,800	17,800
CE-158	ポケコンレベルコンバター	39,800	31,300
CE-159	ポケコンRAM 8K	35,000	4,200
CE-140T	ポケコンRS-232Cコンバター	9,800	8,800
CE-140F	ポケコンフロッピディスク	49,800	44,800
CE-123P	ポケコンプリンター	19,800	17,800
CE-120P	ポケコンプリンター	24,800	21,800
CE-126P	ポケコンプリンター	17,800	13,800
CE-124	ポケコンカセットインター	4,500	3,600
Z-VISIONplus	Z80シュミレータ デバッカー	59,800	51,000
UX-1	ホームコピーファクス	78,000	69,800
PA-9500	ハイパー電子手帳	48,000	特価
CZ-300F	X13"マイクロフロッピー	79,800	9,000
CZ-31FS	300F増設フロッピー	59,800	7,000
CZ-82F	CZ-802C増設フロッピー	59,800	6,000
CZ-501H	X1増設用ハードディスクユニット	258,000	60,000
CZ-6BS1	SCSIボード	29,800	23,800
CZ-6BP1	数値演算ボード	79,800	63,800
CZ-6BU1	ユニバーサルI/Oボード	39,800	33,800
CZ-6BM1	MIDIボード	29,800	23,800
CZ-6BEIB	1M増設RAMボード	28,000	19,500
CZ-6BE1	1M増設RAMボード	35,000	29,500
CZ-6BE2	2M増設RAMボード	79,800	63,800
CZ-6BE4	4M増設RAMボード	138,000	110,400
CZ-6BN1	スキャナーボード	29,800	25,300
CZ-6BF1	RS-232C増設ボード	49,800	42,300
CZ-6SD1	システムラック	44,800	38,000
CZ-6TU	RRGBシステムチューナー	33,100	26,500
CZ-6BG1	X6800GPIBボード	59,800	50,000
CZ-6BC1	X6800FAXボード	79,800	67,800
CZ-6PV1	ビデオプリンター	198,000	158,000
CZ-6BV1	ビデオボード	21,000	16,800
CZ-822C	X1G MODEL30	118,000	39,800
CZ-820C	X1G MODEL10	69,800	16,800
CZ-8BGR2	グラフィックボードX1	14,800	3,000
CZ-8BF1	FDインターフェイス	14,800	11,500
CZ-8BK2	X1 漢字ROM	19,800	16,800
CZ-8BM2	232Cマウスボード	19,800	16,800
CZ-8BE2	320K外部メモリー	29,800	25,300
CZ-8BR1	立体映像セット	29,800	25,300
CZ-8BV2	カラーイメージボート	39,800	32,000
CZ-8BO1	FDインターフェイス	14,800	8,000
CZ-8TM1	X1ソフト付モデムユニット	29,800	5,000
CZ-8TM2	X1ソフト付モデムユニット	49,800	39,800

型名	品名	正価	特価
CZ-8EB3	拡張i/o box	33,800	28,000
CZ-8LM1	232cケーブル	7,200	6,000
CZ-8LM2	232cクロスケーブル	7,200	6,000
CZ-8NJ1	ジョイカード	1,700	1,360
CZ-8NT1	トラックボール	13,800	11,500
CZ-8PK10	24ドット136桁漢字プリンター	99,800	大特価
CZ-8PK7	24ドット80桁漢字プリンター	122,000	59,800
CZ-8PC5BK	48ドット熱転写カラー漢字プリンター	96,800	新発売
CZ-8BS1	X1FM音源ボード	23,800	19,800
CZ-8BK4	X1第2水準ROM	――	5,700
CZ-8NJ2	インテリジェントコントローラー	23,800	18,500
CZ-8NS1	カラーイメージスキャナー	188,000	149,000
AN-S100	アンプ付スピーカー	36,600	29,500
AN-X68	キーボードシリコンカバー	3,500	2,800
AN-X68PRO	キーボードシリコンカバー	3,500	2,800
AN-1508	ディスプレイ15P→8P変換ケーブル	――	1,600
AN-1506	ディスプレイ15P→6P変換ケーブル	――	1,600
HXD040	アイティム40Mハードディスク(ITM)	118,000	89,000
HXD140	40Mハードディスク内蔵用(ITM)	98,000	79,800
CU-14FD	カラーディスプレーアナログ0.31	74,800	49,800
MZ-1D10	12"モノクロディスプレー	41,800	25,000
MZ-1D17	15"CRT mz-5500/6500/2	124,000	59,800
MZ-1E05	MZ-2000 FDインターフェイス	24,500	18,000
MZ-1E08	プリンターI/F 2000/2200/80B	9,000	8,000
MZ-1E11	MZ-6500用 SFD I/F	38,000	25,000
MZ-1E04	MZ-2000 プリンターI/F	10,000	6,000
MZ-1E21	MZ-5500 GP I/F	36,000	12,000
MZ-1E18	MZ2000QD用インターフェイス	9,800	3,000
MZ-1E33	MZ6500パラレルI/F	34,800	28,000
MZ-1E45	MZ6500 232C I/F	50,000	15,000
MZ-1E32	MZ2500 パラレル I/F	30,000	27,000
MZ-1E44	MZ-6500 S-RN I/F	50,000	15,000
MZ-1E22	MZ-5500 GPIB I/F	72,800	25,000
MZ-1E29	RS-232Cインターフェイス 300BT	17,800	9,800
MZ-1E01	MZ-3500 232Cボード	28,000	13,000
MZ-1E14	MZ1500 QD用インターフェイス	9,800	3,000
MZ-1M01	MZ-2000/2200 16ビットボード	78,000	8,000
MZ-1M09	MZ-6500 8082-2演算プロセッサ	82,000	30,000
MZ-1M03	MZ-5500 数値演算	69,000	38,500
MZ-1M12	MZ-2861 8087 演算プロセッサ	90,000	45,000
MZ-80P4B	136桁ドットプリンター	――	48,000
MZ-1P06	ドットプリンター	234,000	45,000
MZ-1P27	水平漢字プリンター	268,000	188,000
MZ-1P28	ドットプリンター漢字80桁	148,000	118,400
MZ-1P10A	24ドットプリンター漢字80桁	245,000	79,000
MZ-1P22	熱転写漢字プリンター	59,800	25,000
MZ-1P29	漢字プリンター136桁	168,000	134,400
MZ-1P30	136桁プリンター	228,000	120,000
MZ-1R01	MZ-2000/2200Gボード	39,800	10,000
MZ-1R10	MZ-5500 漢字ROM付	30,000	9,800
MZ-1R09	MZ-5500 V.RAM	35,000	15,000
MZ-1R06	MZ-5500 増設RAM	45,000	8,000
MZ-1R12	MZ-80B/2000/1500/700 RAM	35,000	8,000
MZ-1R11	MZ-5500 256KRAM	80,000	35,000
MZ-1R36	MZ-28611M増設RAM	45,000	15,000
MZ-1R35	MZ-28611M増設RAM	55,000	19,000
MZ-1R14	MZ-5500 辞書ROM	40,000	22,000
MZ-1R16	MZ-5500 128KRAM	30,000	8,000
MZ-1R27A	MZ-2500VRAM	13,000	10,000
MZ-1R21	漢字ROM	38,000	13,000
MZ-1R24	MZ-1500 辞書ROM	22,000	6,000

型名	品名	正価	特価
MZ-1R32	MZ-6500RAM	80,000	40,000
MZ-1R31	漢字ROM	28,000	20,000
MZ-1R28A	MZ-2500 辞書ROM	13,000	10,000
MZ-1R29A	MZ-1P22第2水準漢字ROM	15,000	12,000
MZ-1S13	MZ-1D17チルトスタンド	12,000	5,000
MZ-1T02	MZ-2200 データーレコーダー	19,800	8,500
MZ-1T03	MZ-5500 データーレコーダー	12,000	8,500
MZ-1U09	MZ-2500 拡張ボード	代品在庫少々有り	
MZ-1V01	パソコン FAX	278,000	85,000
MZ-1X22	モデムユニット	21,800	13,000
MZ-2Z016	MZ-5500 附属	――	5,000
MZ-2Z023	MZ-5500 GWBASIC	50,000	30,000
MZ-2Z031	MZ-6500 日本語ワープロ	49,800	15,000
MZ-2Z029	MZ-6500 TODAY	68,000	20,000
MZ-2Z064	MZ-6500 書院RAM付	69,800	28,000
MZ-2Z065	MZ-6500 書院RAMなし	49,800	15,000
MZ-2Z012	MZ-5500 附属	――	5,000
MZ-2Z013	MZ-5500 MSDOS	25,000	20,000
MZ-4Z001	MZ-5500 IBM変換	30,000	8,000
MZ-5521	本体	388,000	55,000
MZ-5511	本体	288,000	35,000
MZ-5Z013	MZ-1500 QD通信ソフト	――	3,500
MZ-6F03	ブランク QD DISK	450	400
MZ-6P18	MZ-1P18,28カットシートフィーダー	60,000	35,000
MZ-6P29	MZ-1P29 カットシートフィーダー	50,000	37,500
MZ-6P27	MZ-1P27 カットシートフィーダー	58,000	39,800
MZ-6P06	MZ-1P06 トラクターフィード	15,000	7,500
MZ-6P20	MZ-1P22/17ロールホルダー	3,100	2,700
MZ-6Z22	MZ-6500(50)CP/M86BASIC-3	10,000	6,000
MZ-6Z25	M-50 ストリーマユーティリティZプロセッサ	39,800	15,000
MZ-80T20A	MZ-80 マシンランゲージ	6,000	5,000
MZ-80TUB	MZ-80 バックアップ	20,000	8,000
MZ-80TU	MZ-80 システムプログラム	20,000	8,000
MZ-80T40A	MZ-80 PASCAL	10,000	5,000
MZ-80T70A	MZ-80 FDOS	20,000	7,000
MZ-8BGK	MZ-80 BGRAM2	39,000	10,000
MZ-8B104	MZ200/2200 GP IBインターフェイス	45,000	18,000
MZ-8BC01	MZ200/2200 GP IBケーブル	18,000	8,000
UE-1U01	X286L スロットBOX	5,000	4,000
UE-1R02	4M RAMボード	300,000	240,000
UE-1R06	辞書ROMボート	32,800	25,600
UE-1R01	2M RAMボード	160,000	128,000
UE-1R05	拡張グラフィックボード	92,000	55,000
UE-1R03	1M RAMボード	100,000	80,000
UE-1R04	2M RAMボード	180,000	144,000
UE-1P03	80桁漢字プリンタ	――	特価
UE-1P04	136桁漢字プリンタ	――	特価
UE-1P05	136桁漢字水平プリンタ	――	特価
UE-1P02	高速136桁漢字プリンタ	550,000	440,000
UE-1P01	136桁漢字プリンタ	268,000	214,400
UE-1E04	S-RNインターフェイスカード	70,000	56,000
UE-1E02	AX286L ICカードI	45,000	36,000
UE-1E03	5"FDインターフェイスカード	28,000	22,400
UE-1D03	15インチカラーディスプレイ	123,000	98,400
UE-1D02	14インチカラーディスプレイ	158,000	126,400
IO-735X	カラープリンター	248,000	180,000
BF-68PRO	フィルター	19,800	16,800
X68000キーボード延長ケーブル(1.5m)		2,500	2,000
ディスプレーケーブルアナログ15P(3m)		5,000	4,000
ディスプレーケーブルアナログ15P(1.5m)		4,300	3,500

ポケコン関係周辺機器サプライ製品及シャープ関係のソフトウエア全種取扱います。

FM TOWNS/FM NOTE/東芝ダイナブック、周辺機器も取扱っております。

XVI発売記念X68000F下取りセール！

XVI CZ-634C-TN
標準価格 ￥368,000
XVI CZ-644C-TN
標準価格 ￥518,000

SHARP X68000シリーズ対応 ハードディスク

〈ITEM〉

HXD 040 23mS X68000
定価￥118,000⇨特価￥79,800

HXD 042 X68000 増設用
定価￥128,000⇨特価￥102,500

HXD 140 X68000 内蔵用
定価￥98,000⇨特価￥79,800

HXD-140は602C、603Cの内蔵用

X68000

CZ-604C プラス（ディスプレイ）組合せ

CZ-606D	￥290,000
CZ-602D	￥305,000
CZ-604D	￥300,000
CZ-613D	￥330,000

CZ-602C（本体） プラス（ディスプレイ）組合せ

CZ-606D	￥270,000
CZ-613DGY	￥310,000
CZ-605DGY	￥300,000
CZ-611DGY	￥285,000

CZ-602C（本体 40Mハードディスク付） プラス（ディスプレイ）組合せ

CZ-603DGY	￥305,000
CZ-602D	￥340,000
CZ-612D	￥345,000
CZ-613D	￥365,000

CZ-652C（本体） プラス（ディスプレイ）組合せ

基本セット ￥228,000

CZ-603C（本体 40Mハードディスク付） プラス（ディスプレイ）組合せ

CZ-603DGY	￥365,000
CZ-602D	￥380,000
CZ-612D	￥385,000
CZ-613D	￥400,000

CZ-603CBK（本体） プラス（ディスプレイ）組合せ

CZ-606D	￥270,000
CZ-602D	￥285,000
CZ-604D	￥280,000
CZ-613D	￥310,000

CZ-653C（本体） プラス（ディスプレイ）組合せ

基本セット ￥248,000

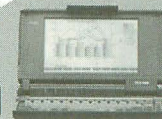
富士通関係在庫一覧（新品）

品名	型名
（本体関係）	
FM-New7	MB25015
FM77-L4	MB25260
FM77AV-1	FM77AV-1
FM77AV-2	FM77AV-2
FM77AV20-1	FM77AV20-1
FM77AV20-2	FM77AV20-2
FM77AV40	FM77AV40
FM77AV20EX	FM77AV20EX
FM77AV40EX	FM77AV40EX
FM-16βFD	MB25420
FM-16βSD	MB25410
FM-16βFD-I	FM16B-FD1
FM-16βFD-II	FM16B-FD2
FM-16βHD-I	FM16B-HD1
FM-16βHD-II	FM16B-HD2
FM-16π	MB25320
FM-16π	MB25321

品名	型名
((CRT関係)	
カラーCRTディスプレイ	MB27331
カラーCRTディスプレイ	MB27333
カラーCRTテレビ	FMTV-151
カラーCRTテレビ	FMTV-152
カラーCRTテレビ	FMTV-153
カラーCRTテレビ	FMDPC231D
（プリンター、その他）	
日本語カード	fm77-211
MIDアダプター	FM77-401
熱転写プリンター	MB-27407
漢字熱転写	MB-27413
漢字	MB-27410E
漢字	MB27410A
シリアルドット	MB27409
漢字	MB27411E
カラー漢字	FMPR-451
漢字	FMPR-353
カラー熱転写	FMPR-201
カラー漢字熱てん	PR-203W2
漢字	FMPR-301

価格については、資料ご請求下さい。

TOSHIBA J-3100SS001 DynaBook

純正キャリングケースプレゼント

定価￥198,000 ➡特価￥99,800

富士通FM TOWNSお買得セット

FM TOWNS TOWNSモデル2基本セット

TOWNS-2	￥398,000
FMT-DP533	￥69,800
FMT-KB101	￥20,000
B276A010(V.3)	￥20,000
定価合計	￥507,800
特価	**￥228,000**

FM TOWNS TOWNSモデル20F基本拡張セット+MS-DOS

TOWNS20F	￥323,000
FMT-DP533	￥69,800
FMT-KB105/205	￥30,000
MS-DOS(V.3.1)	￥18,000
FM秘書	￥20,000
定価合計	￥460,800
特価	**￥328,000**

FM TOWNS TOWNSモデル40H基本セット+MS-DOS

TOWNS40H
FMT-DP533
FMT-KB105/205
MS-DOS(V.3.1)
FM秘書

特価 ￥458,000

FM TOWNS TOWNSモデル80H基本セット+MS-DOS

TOWNS80H
FMT-DP533
FMT-KB105/205
MS-DOS(V.3.1)
FM秘書

特価 ￥571,000

40H、80H等、その他の組合せもご相談下さい。

アイビット推奨ディスプレイ

シャープ
CZ-612DGY
ドットピッチ0.31
チルト台付
特価￥80,000

CZ-880D/860Dの代品
シャープ
CU-14TV
ドットピッチ0.31
￥64,800

シャープ
CZ-602D-BK
（15型アナログTV/
3モードオートスキャン）
特価￥75,000

FMTV-154 ￥129,200→**￥75,000**
15型デュアルスキャン15K/アナログ21PTVチューナー付

FMD-PC231D ￥89,800→**￥45,000**
15型デュアルスキャン15K/24Kアナログ21P

※富士通、NEC、シャープ周辺機器（拡張機器全機種、プリンター他）も常時取り扱っております。

〈全商品新品完全保証付〉シャープ、カシオポケコン全機種取扱い、カタログ、価格表ご請求には、72円を添えてお願い致します。

☎0426-45-3002（駅前）-3001（本店）-3003（教室）
FAX.0426-44-6002

●営業時間/10:00～19:00●電話受付/20:00迄可●定休日/水曜日

SHARP SUPER XEX SHOP

アイビット電子株式会社 〒192 東京都八王子市北野町560-5

上記の広告商品はすべて店頭販売もしております。

★送料はご注文の際にお問い合わせ下さい。
★掲載の商品は、すべて新品、保証書付きです。
★掲載の商品は充分用意してありますが、ご注文の際は、在庫の確認の上、現金書留または、銀行振込でお申し込み下さい。全商品クレジットでも扱っております。
★お申し込みの際は必ず電話番号を明記して下さい。
★商品、品切れの節はご容赦下さい。

富士銀行八王子支店 （普）1752505

●本誌発売時には上記価格よりさらにお求めやすい価格に変更されている場合があります。●この広告の商品にはすべて送料・消費税は含まれておりません。

OAB特選〜X68000シリーズセット　★もし！他店より高い場合はTEL下さい!!

①X68000XVI
NEW
●SX-WINDOW搭載!!

- CZ-634C-TN
- CZ-613D-TN
- MD-2HD 20枚

定価合計￥503,000

特価 ￥TEL下さい!!

新製品発売記念に付先着10名様に数値演算プロセッサプレゼント!!

②X68000XVI-HD
- CZ-644C-TN
- CZ-613D-TN
- MD-2HD 20枚

定価合計￥653,000

特価 ￥TEL下さい!!

③X68000 PROII
- CZ-653C-BK/GY
- CZ-605D-BK/GY
- MD-2HD 20枚

定価合計￥400,000

特価 ￥

●SX-WINDOW搭載!!

④X68000PRO II-HD
- CZ-663C-BK/GY
- CZ-605D-BK/GY
- MD-2HD 20枚

定価合計￥510,000

特価

★X68000XVIお買上げの方全員にプレゼント①〜④の中から選んで下さい!!

①PA-9500（電子システム手帳）定価￥48,000　②MD-24FS5（モデム）定価￥49,800　③スーパーファミコン（ゲームソフト3本付）　④CJ-S220（コードレスホン）定価￥44,000

X68000 SUPER-HD
●SX-WINDOW搭載!!

- SX-WINDOW搭載
- SCSIインターフェース装備
- 80MBハードディスク搭載
- 3MB大容量メモリ装備
- 高解像度グラフィック

⑤X68000 SUPER-HD
台数限定
- CZ-623C-TN（チタン）
- CZ-613D-TN（チタン）
- MD-2HD 20枚

定価合計￥633,000

特価

X68000 特選OABセット★本体のみ単品OK!!
① CZ-604C-TN（新品）＋CZ-606D-BK（新同品）　3セット限り………特価

② CZ-602C-BK（新品）＋CZ-606D-BK（新同品）　1セット限り………特価￥218,000

③ CZ-603C-GY（新品）＋CZ-606D-GY（新同品）　3セット限り………特価￥238,000

④ CZ-652C-GY（新品）＋CZ-606D-GY（新同品）　2セット限り………特価￥199,000

オーエーブレイン　全国通販
●オフコンからパソコンまで 幅広〜い品揃え。おまかせあれ!! お電話くださいネ！ ☎03-5688-3621

- ★全商品保証書付。専門のアドバイザーがお客様のニーズに親切に対応します。
- ★初期不良・輸送トラブル等に迅速に対応し、即交換させていただきます。
- ★送料は1個につき￥1,000です。（※一部離島は除きます。お問合せ下さい。）
- ●ご注文、お問合せは…毎日午前10時から午後8時まで
- ●下取・買取は電話で見積りしております。責任を持って下取りさせて頂き
- ●商品のお届けは…入金確認後、即日発送致します。

周辺機器コーナー

プリンターセットコーナー
- ●CZ-6PVI（カラービデオプリンター）定価￥198,000………▶特価￥148,000
- ●CZ-8PC3（24ドット熱転写カラープリンター）定価￥65,800………▶特価￥53,000
- ●CZ-8PK10（24ピン漢字ドットプリンター・136桁）定価￥97,800………▶特価￥71,000
- ●CZ-8PGI（24ピンカラー漢字ドットプリンター・80桁）定価￥130,000………▶特価￥92,000
- ●CZ-8PG2（24ピンカラー漢字ドットプリンター・136桁）定価￥160,000………▶特価￥114,000
- ●IO-735X（カラーイメージェットプリンター）定価￥248,000………▶特価￥178,000

X68000用ソフトウェアー・コーナー
ソフト	定価	特価
①CZ-212BS(BUSINESS)	￥68,000	￥53,000
②CZ-220BS(DATA)	￥58,000	￥45,000
③CZ-215MS(Sampling)	￥17,800	￥13,800
④CZ-221HS(NEW Print Shop)	￥10,800	￥15,500
⑤CZ-227BS(TOP財務会計)	￥200,000	￥158,000
⑥CZ-226BS(CARD)	￥229,800	￥23,000
⑦CZ-223CS(Communicabion)	￥19,800	￥115,500
⑧CZ-213MS(MUSIC)	￥18,800	￥14,800
⑨CZ-211LS(C compiler)	￥39,800	￥31,000
⑩C-TRACE(キャスト)	￥68,000	￥52,000
⑪EW(イースト)	￥38,000	￥29,000

特選品
■CZ-8PC5（48ドット熱転写カラー漢字プリンター）（定価￥96,800）……特価￥71,500

X68000用周辺機器コーナー
型番	品名	定価	特価
CZ-6BEI	IBM増設RAMボード	￥35,000	￥25,500
CZ-6BEIB	IBM増設RAMボード	￥28,000	￥20,500
CZ-6BE2	2MB増設RAMボード	￥79,800	￥59,000
CZ-6BE4	4MB増設RAMボード	￥138,000	￥102,500
CZ-6BF1	増設用RS-232Cボード	￥49,800	￥37,000
CZ-6BGI	GP-IBボード	￥59,800	￥43,500
CZ-6BMI	MIDIボード	￥26,800	￥19,500
CZ-6BNI	スキャナ用パラレルボード	￥29,800	￥22,000
CZ-6BPI	数値演算プロセッサボード	￥79,800	￥59,000
CZ-6BOI	ユニバーサルI/Oボード	￥39,800	￥29,500
CZ-6EBI/BK	拡張I/Oボックス	￥88,000	￥64,000
CZ-6VTI/BK	カラー・イメージ・ユニット	￥69,800	￥51,000
CZ-8NM24	マウス	￥6,800	￥5,000
CZ-8NTI	マウストラックボール	￥9,800	￥7,000
CZ-8NSI	カラーイメージスキャナ	￥188,000	￥135,000
CZ-6BCI	FAXボード	￥79,800	￥59,000
CZ-8TM2	モデムユニット	￥49,800	￥37,000
CZ-64H	増設ハードディスク	￥120,000	￥87,000
CZ-6TU GY/BK	RGBシステムチューナー	￥33,100	￥24,000
BF-68PRO	高性能CRTフィルタ	￥19,800	￥15,000
CZ-6MOI	光磁気ディスクユニット	￥450,000	￥327,000
CZ-6BSI	SCSIインターフェースボード	￥29,800	￥22,000
CZ-6BL2	LANボード	￥298,000	￥217,000

I·O DATA 増設RAMボード　限定
- ●1MB増設PAMボード PIO-6BEI-A　定価￥25,000　特価￥16,800
- ●2MB増設RAMボード PIO-6BE2-2M　定価￥50,000　特価￥34,800
- ●4MB増設RAMボード PIO-6BE4-4M　定価￥88,000　特価￥59,800

ハードディスク　★その他特価品有！TEL下さい!!
- ■シャープ CZ-64H……特価￥86,000
- CZ-68H……特価￥118,000
- ■ロジテック LHD-200…特価￥218,000
- ■アイテム HXD-040…特価￥88,000
- HXD-042…特価￥95,000
- ■アイテック ●ITX-640‥特価￥84,000
- ●ITX-680‥特価￥99,000
- ●ITX-80S·特価￥87,500
- ●ITX-130S 特価￥106,000
- ★SCSIボード………特価￥22,000

中古パソコン
機種	価格
PC-9801RA2	￥248,000より
PC-9801RX2	￥180,000より
PC-9801VX21	￥175,000より
PC-9801VM21	￥140,000より
PC-9801VM2	￥125,000より
PC-9801F2	￥48,000より
PC-9801EX2	￥180,000より
PC-9801UV21	￥115,000より
PC-9801LV21	￥143,000より
PC-286V	￥125,000より
PC-286VE	￥130,000より
PC-286L	￥110,000より
PC-286LS	￥220,000より
PC-8801FH	￥48,000より
PC-8801MA	￥55,000より
X68000	￥140,000より
X68000(HD)	￥190,000より
X1ターボZII	￥58,000より
FM77AV40EX	￥45,000より
200ラインCRT	￥8,000より
400ラインCRT	￥30,000より
80桁プリンタ	￥15,000より
135桁プリンタ	￥35,000より

●その他多数有り、お問い合せ下さい。

ユニット
- ●FD-1155D（5インチ）……￥9,0
- ●FD-1155C（5インチ）……￥8,0
- ●FD-1165A（8インチ）……￥2,0
- ●FD-1137D（3.5インチ）…￥9,0
- ●D-5146H（5インチ40MB）‥￥34,0
- ●D-3142（3.5インチ40MB）·￥35,0
- ●D-3148（3.5インチSISC）…￥37,0
- ●外付8インチ2ドライブ……￥20,0
- ●外付5インチ2ドライブ……￥30,0

通信販売によるご購入方法（お電話でお申し込み下さい。）

現金一括払い
銀行振込：電信扱いにてお振込下さい。手数料はお客様負担となります。

現金書留：住所、氏名、電話番号、商品名、使用機種、メディア等をお書き添えのうえ、現金書留にて当社までお送り下さい。

クレジット
専用のお申し込み用紙をお送り致しますので、必要事項をご記入・捺印のうえ、ご返送下さい。

※未成年者の方は、保護者のご承認を受けてからお申し込み下さい。

振込先
- ●第一勧業銀行　御徒町支店（普）1376679　オーエーブレイン
- ●朝日信用金庫　本店（普）334833　オーエーブレイン

★クレジットは1〜60回払いで月々5,000円よりご自由に設定できます。

オーエーブレイン　〒110 東京都台東区台東1-28-4　TEL & FAX 5688-3621

オーエーブレイン今月の特価品!! 台数限定 お早目に!!

ドライブ・ユニット
コンピュータ・リサーチ（自動切換）
- ●CRC-FD3.5S…特価￥25,000
- ●CRC-FD3.5W…特価￥38,000
- ●CRC-FD 5S……特価￥30,000
- ●CRC-FD 5W……特価￥45,000
- ●CRC-FD 5N……特価￥32,000

グローリア（1MB専用）
- ●GD-35MI………特価￥22,000
- ●GD-35M2………特価￥39,000
- ●GD-50MI………特価￥26,000

緑電子（1MB専用）
- ●Little-F1………特価￥24,000
- ●Little-F2………特価￥36,000

ハードディスク 内蔵
■コンピュータ・リサーチ
- ●CRC-IHR4（40M）（定価￥98,000）………特価￥58,000
- ●CRC-IHR8/E8（80M）（定価￥158,000）……特価￥80,000
- ●CRC-MH8B（80M）（定価￥158,000）………特価￥80,000

サウンド・ボード
SNE
- ①サウンドオーケストラV……特価￥23,000
- ②サウンドオーケストラL……特価￥17,800
- ③リトルオーケストラ………特価￥14,500
- ④リトルオーケストラL……特価￥12,000
- ⑤サウンド・ミュージシャン‥特価￥16,000
- ⑥リトルミュージシャン……特価￥8,500

ハード・ディスク（外付
■コンピュータ・リサーチ
- ●CRC-MH8B（80M）………特価￥
- ●CRC-MH4B（40M）………特価￥

■緑電子
- ●LITTLE-E40（40M）………特価￥

プリンター
■キャノン
- ●BJ-10V（￥74,800 48ドットバブルジェット‥特価￥
- ●LBP-B406S＋トナ ………特価￥3
- ●LBP-A404＋トナー ………特価￥1

■流通事情により、広告表示よりお安くなる場合もございます。まずは、お電話下さい。■ビジネス・ゲームセットもございます。

ACCESS

X1エミュレータ

好評発売中

定価¥9,800

X1エミュレータはX68000上でX1シリーズのアプリケーションを実行するためのソフトエミュレータです。X1のアプリケーションを完全にソフトウェアのみでエミュレートしているため、X1上での実行速度と比較して、平均3〜5倍程度おそくなりますが、X68000のマシン上に実現した仮想X1マシンを楽しめます。また、X1とX68000の相互間でファイルを転送するためのユーティリティと専用ケーブルが付属しますので、X1上で作り上げたソフトの資産をX68000上に移行することも簡単にできます。

X1エミュレータの機能

- X1エミュレータはX1に相当する機能をエミュレート。
  この仮想コンピュータには最大4つのドライブが仮想的に接続。
- X1エミュレータからみたドライブはHuman68kのドライブ上にあるファイルで仮想的に実現。このファイルはX1用の5″2Dディスクのイメージをファイル転送ユーティリティでまるごと転送したもの。
- X1エミュレータで仮想的に実現したX1は仮想ドライブから起動。
  このため仮想ドライブ用ファイルには、X1を立ち上げるために必要なHuBASICやCP/Mなどのシステムプログラムが必要。
- X1エミュレータでは、X1の持つVRAMを含むメモリイメージやZ80CPUを仮想的にソフトウェアで実現。

ファイル転送ユーティリティ

ディスク転送

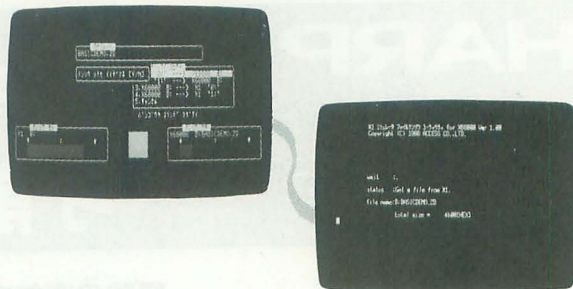
X1ディスク⟷X68000 Human68k(5″2Dディスクイメージファイル)

- X1エミュレータではHuman68k上のディスクイメージファイルを仮想ドライブとして使用。

ファイル転送

X1 BASIC : CP/M⟷X68000 Human68k

- X1で作ったプログラム&データをX68000上で使用。

※付属の専用ケーブルをX1とX68000に接続してファイルを転送します。

X1エミュレータ Q&A

Q. ファイル転送のために別途RS-232Cケーブルを買わないといけないのですか?

A. 専用のケーブルが付属しますのでその必要はありません。

Q. X1BASICのプログラムをX68000上のX-BASICで使えますか?

A. 通常のセーブではコードが違うので使用できませんが、アスキーセーブしたファイルであればX-BASIC上でそのままロード可能です。

Q. TurboBASICで作成した住所録などの漢字を含んだデータがあるのですがX68000上にファイル転送できますか?

A. X1TurboもX68000も漢字はシフトJISコードなのでファイルの転送は可能です。ただし、漢字ROMを必要とするものはサポートしていません。

Q. Turbo用のソフトは動きますか?

A. X1用のみでTurbo専用のソフトは動きません。

Q. ゲームは動きますか?

A. 純粋にBASICでかかれたものは動きますが、プロテクトがかかったものや直接ハードをアクセスするような市販のゲームは動きません。

＊タイミング等ハードウェアに依存するようなソフトは、原理上実行できない、もしくは正常に動作しない場合がありますのでご注意ください。
＊一部サポートしていない機能があります。

X1エミュレータ通信販売 購入希望として住所、氏名、電話番号をお知らせください。注文書をお送り致します。

＊この商品価格には消費税は含まれておりません。
＊CP/Mはデジタルリサーチ社の商標です。
文中のソフトウェアは各社の商標です。
＊製品の仕様、名称は予告なく変更する場合もございますのであらかじめご了承ください。

有限会社アクセス
〒101 東京都千代田区神田神保町1-64
神保町協和ビル7F
☎03(3233)0200(代) FAX.03(3291)7019

資料請求券 oh!X

パソコン/ワープロ通信ネットワークサービス

J&P HOT LINE NEWS

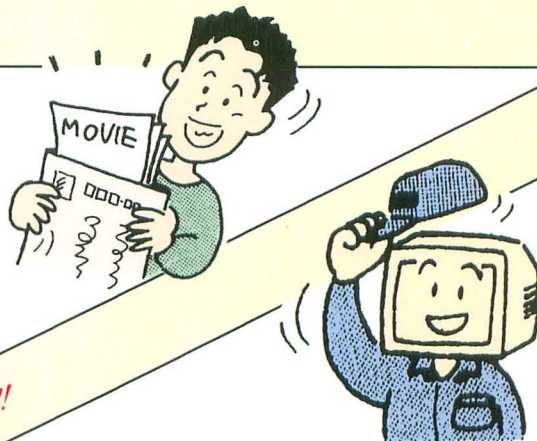
Time is money!

時間の有効活用をお手伝いする新サービス登場!

カタログショッピング(ジャンプコード:CATALOG)

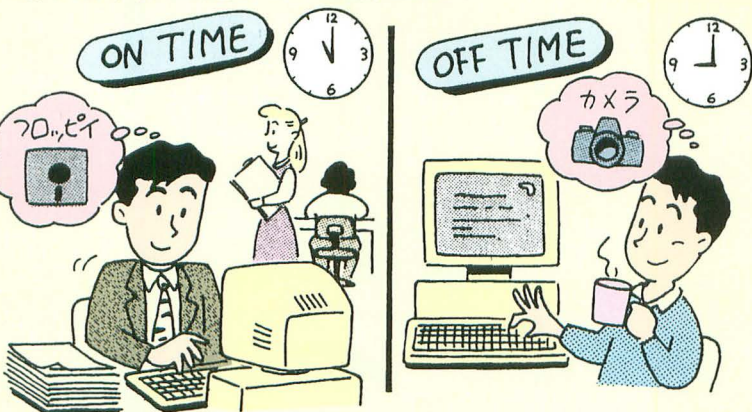
面倒なカタログ集めもこれで大丈夫!

カタログショッピングといっても、専用カタログがあるわけではありません。「あの商品が欲しいな」「こんな商品があったらいいな」といった欲しい商品、探されている商品のカタログをお送りするというものです。更に、HOT LINE特別斡旋割引にて購入できるので、ものは試し、一度申し込んでみてください。

(カタログ送付だけの場合は、送付手数料として200円を申し受けさせていただきます。)

日常的によく使う消耗品で、いちいち商品を見比べて買わなくても「これ」と決まっている商品――たとえばプリンタのインクリボンや用紙、ディスケットなどのOAサプライ。わざわざ出かけて買いにいくのは意外と手間なもの。これらをあなたに代わって手配し、お届けしようというものです。申し込み方法も簡単!必要なものを電子メールでオーダーするだけ。時間を有効に活用される方にぜひご利用いただきたいサービスです。

※詳しくはネット上の各コーナーをご覧ください。

ON TIME

フロッピイ

その他 楽しいメニューがまだまだいっぱい!

- ★J&Pならではのパソコン・家電製品の会員割引もある**ONLINE SHOPPING**。
- ★J&Pだから強い!!パソコン情報をはじめとする役に立つ**DATA BASE**。
- ★みんなでおしゃべり**オンライントーク**(CHAT機能)。
- ★地域別・テーマ別ボードで充実の**BBS**(電子掲示板)。
- ★ビジュアルデータもばっちり送受信できる**X-MODEM**。

J&P HOT LINEへのご入会はスタータキットで。

お求めは、下記のお店へ。又は現金書留にて、¥3,000+¥90(消費税3%)=¥3,090を事務局までお送り下さい。
すぐにスタータキットをお送りします。

お問い合わせは 〒556 大阪市浪速区日本橋西1-6-5 上新電機株式会社
J&P HOT LINE事務局宛 TEL.(06)632-2521

スタータキットのお求めはJ&P各店でどうぞ。

店名	住所	電話
渋谷店	東京都渋谷区道玄坂2丁目28番4号	☎(03)3496-4141
町田店	東京都町田市森野1丁目39番16号	☎(0427)23-1313
八王子店	東京都八王子市旭町1番1号八王子そごう7F	☎(0426)26-4141
立川店	東京都立川市幸町4-39-1	☎(0425)36-4141
本厚木店	厚木市中町3-4-3	☎(0462)25-1548
富山店	富山市桜町2-1-10	☎(0764)32-3133
金沢店	金沢市入江2-63	☎(0762)91-1130
寺地店	金沢市寺地2-3	☎(0762)47-2524
大須店	名古屋市中区大須4丁目2-48	☎(052)262-1141
テクノランド	大阪市浪速区日本橋5丁目6番7号	☎(06)634-1211
メディアランド	大阪市浪速区日本橋5丁目8番26号	☎(06)634-1511
コスモランド	大阪市浪速区難波中2丁目1番17号	☎(06)634-3111
U. S. LAND	大阪市浪速区日本橋4丁目9番15号	☎(06)634-1411
ビジネスランド	大阪市北区梅田1-1-3大阪駅前第3ビルB2	☎(06)348-1881
梅田店	大阪市北区小松原町1-10	☎(06)362-1141
高槻店	高槻市高槻町11番16号	☎(0726)85-1212
くずは店	枚方市楠葉花園町15番2号	☎(0720)56-8181
千里中央店	豊中市新千里東町1-3 SENCHU PAL 2番街4F	☎(06)834-4141
摂津富田店	高槻市大畑町24-10	☎(0726)93-7521
寝屋川店	寝屋川市緑町4-20	☎(0720)34-1166
藤井寺店	藤井寺市岡2丁目1番33号	☎(0729)38-2111
岸和田店	岸和田市土生町2451-3	☎(0724)37-1021
さんのみや1ばん館	神戸市中央区八幡通3-2-16	☎(078)231-2111
西宮店	兵庫県西宮市河原町5-11	☎(0798)71-1171
姫路店	姫路市東延末1丁目1番住友生命姫路南ビル1F	☎(0792)22-1221
京都寺町店	京都市下京区寺町通仏光寺下ル恵比須之町549	☎(075)341-3571
京都近鉄店	京都市下京区烏丸通七条下ル東塩小路町702	☎(075)341-5769
和歌山店	和歌山市元寺町4丁目4番地	☎(0734)28-1441
奈良1ばん館	奈良市三条町478-1	☎(0742)27-1111
郡山インター店	大和郡山市横田693-1	☎(07435)9-2221
熊本店	熊本市手取本町4-12	☎(096)359-7800

T4910217906605 雑誌 02179-6